Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 4.2: Aligned Representations

Louis-Philippe Morency

** Co-lecturer: Paul Liang. Original course co-developed with Tadas Baltrusaitis. Spring 2021 and 2022 editions taught by Yonatan Bisk.*

Administrative Stuff

First Project Assignment

Due date: Sunday 9/24 at 8pm

Four main sections:

- Introduction
- Related work
- Experimental setup
- Research ideas

The two main sections are related work and research ideas

teammates = # research ideas

Page limit depends on team size:

- 3 students : 4 pages + references
- 4 students : 4.5 pages + references
- 5 students : 5 pages + references

Follows ICML paper format

Team Meetings with Instructor

- No lecture on Tuesday 10/3
- 15-mins meeting with instructor
  - Optional, but highly suggested
  - Not all teammates are required to attend
- Meetings next week: Wednesday 9/27 and Friday 9/29
- Signup form: https://calendly.com/morency/student-meetings

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 4.2: Aligned Representations

Louis-Philippe Morency

** Co-lecturer: Paul Liang. Original course co-developed with Tadas Baltrusaitis. Spring 2021 and 2022 editions taught by Yonatan Bisk.*

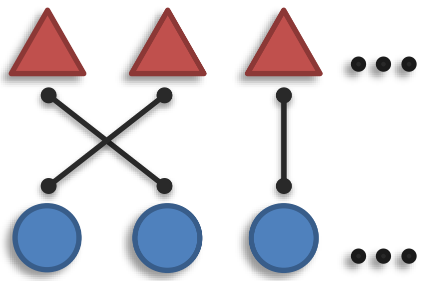
Continuous Alignment

Challenge 2: Alignment

Definition: Identifying and modeling cross-modal connections between all elements of multiple modalities, building from the data structure

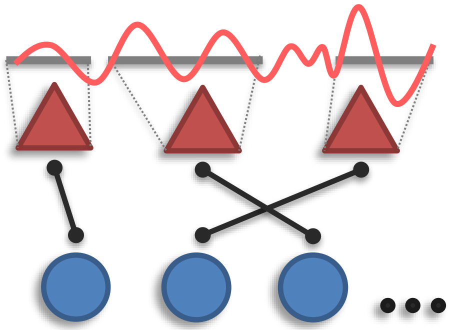
Sub-challenges:

Discrete Alignment



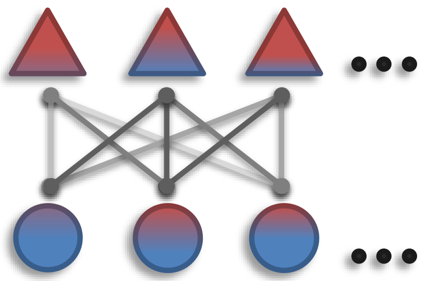
Discrete elements and connections

Continuous Alignment



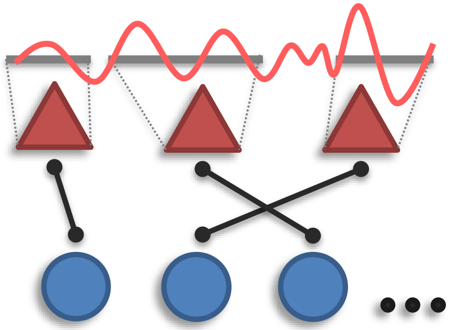
Segmentation and continuous warping

Contextualized Representation



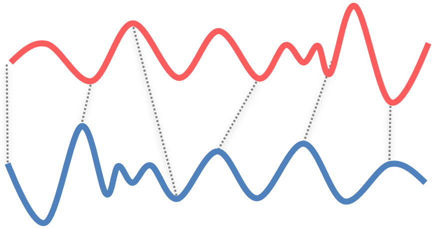
Alignment + representation

Challenge 2b: Continuous Alignment

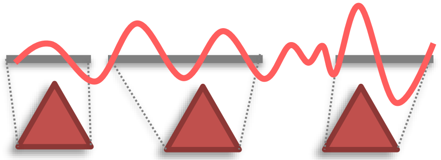


Definition: Model alignment between modalities with continuous signals and no explicit elements

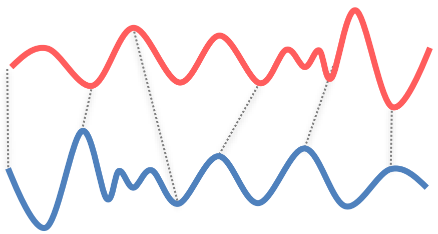
Continuous warping



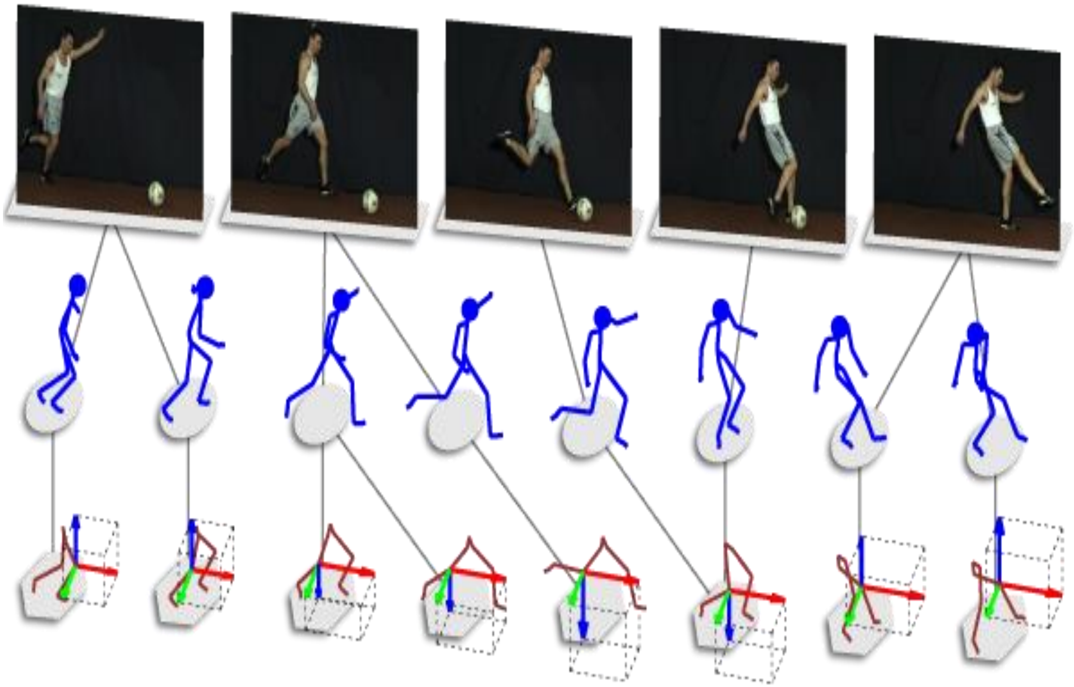
Discretization (segmentation)



Continuous Warping – Example



Aligning video sequences



Dynamic Time Warping (DTW)

We have two unaligned temporal unimodal signals

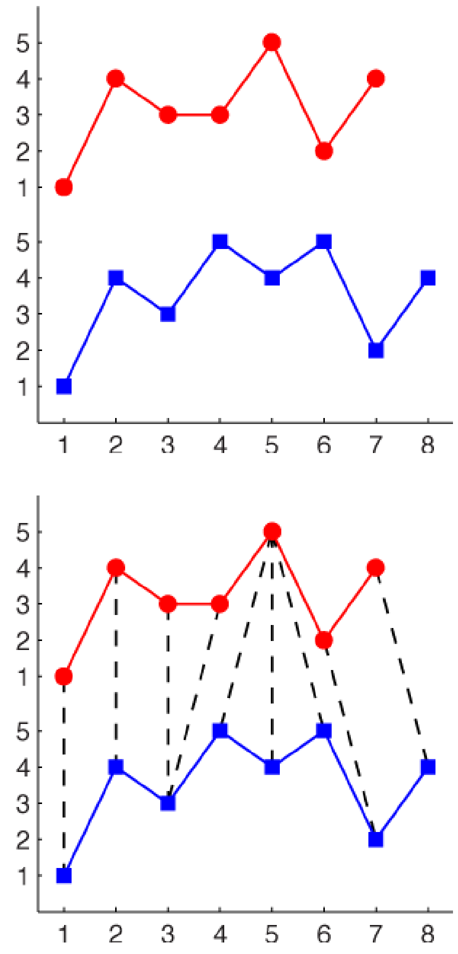
- $\mathbf{X} = \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{n_x}\right] \in \mathbb{R}^{d \times n_x}$
- $\mathbf{Y} = \left[\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_{n_y}\right] \in \mathbb{R}^{d \times n_y}$

Find set of indices to minimize the alignment difference:

$$L(\boldsymbol{p}^x, \boldsymbol{p}^y) = \sum_{t=1}^{l} \left\| \boldsymbol{x}_{\boldsymbol{p}_t^x} - \boldsymbol{y}_{\boldsymbol{p}_t^y} \right\|_2^2$$

where $\boldsymbol{p}^x$ and $\boldsymbol{p}^y$ are index vectors of same length

Dynamic Time Warping is designed to find these index vectors!

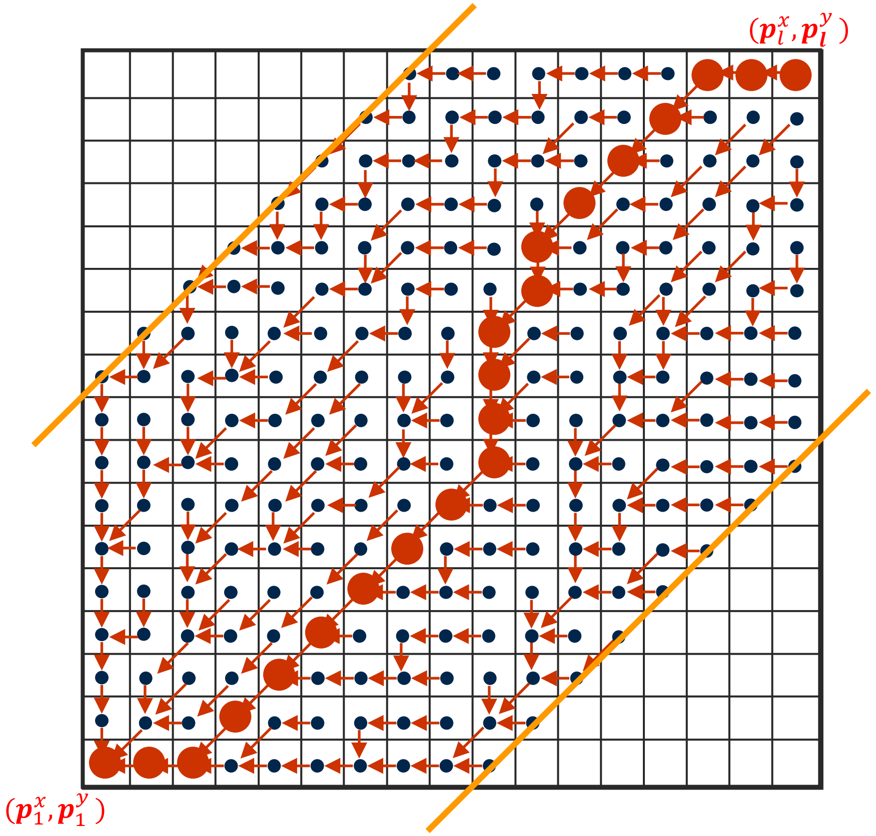


Dynamic Time Warping (DTW)

Lowest cost path in a cost matrix

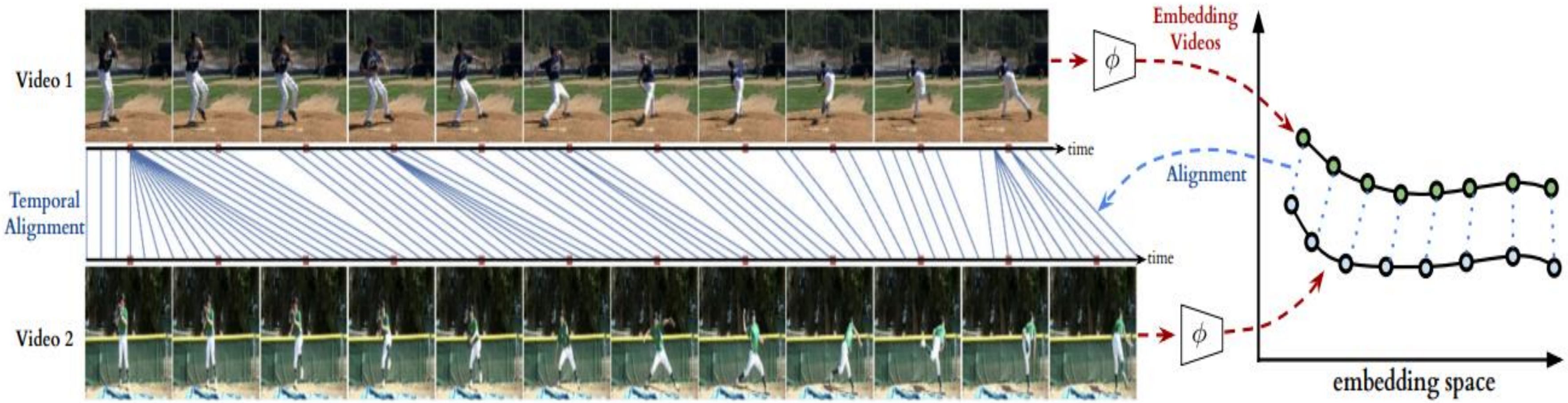
- Restrictions?
  - Monotonicity – no going back in time
  - Continuity - no gaps
  - Boundary conditions - start and end at the same points
  - Warping window - don't get too far from diagonal
  - Slope constraint – do not insert or skip too much

Solved using dynamic programming while respecting the restrictions



Temporal Alignment and Neural Representation Learning

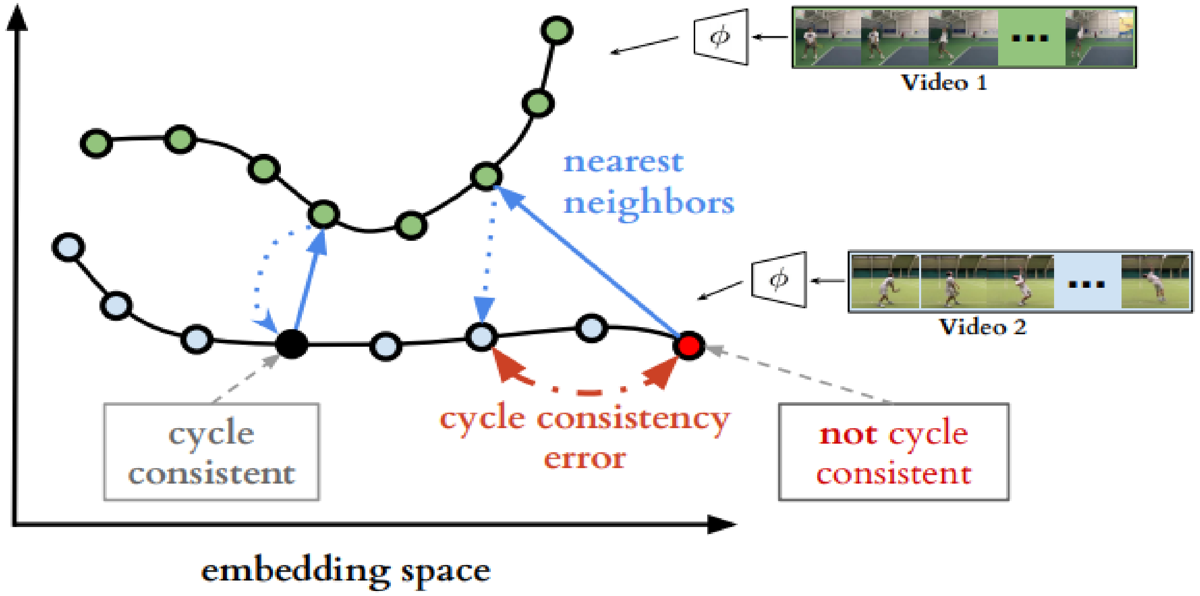
Premise: we have paired video sequences that can be be temporally aligned



How can we define a loss function to enforce the alignment between sequences while at the same time learning good representations?

Temporal Cycle-Consistency Learning

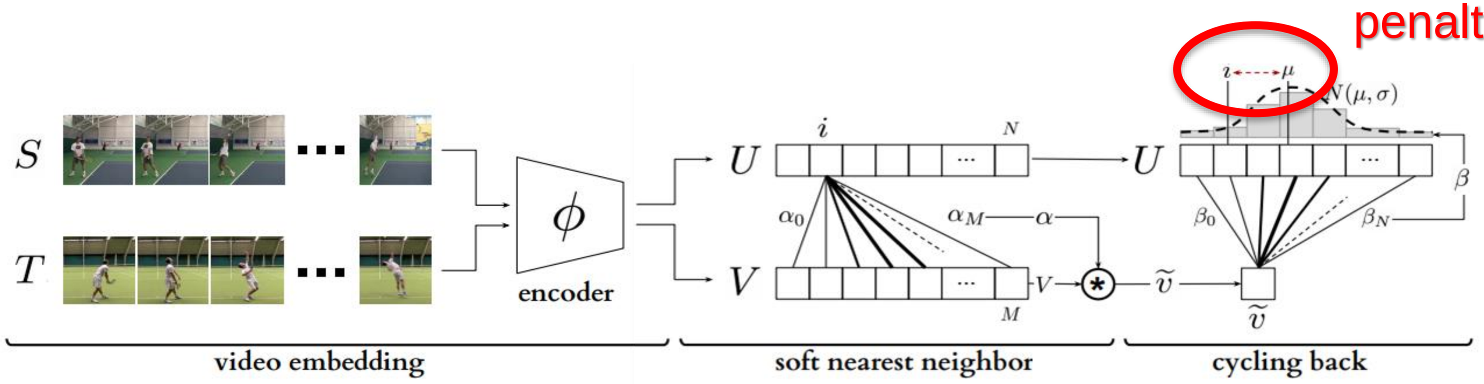
Solution: Representation learning by enforcing **Cycle consistency**



Main idea: My closest neighbor also views me as their closest neighbor

Temporal Cycle-Consistency Learning

penalty!



Compute “soft” / “weighted” nearest neighbour:

distances: $$\alpha_j = \frac{e^{-||u_i - v_j||^2}}{\sum_k^M e^{-||u_i - v_k||^2}}$$

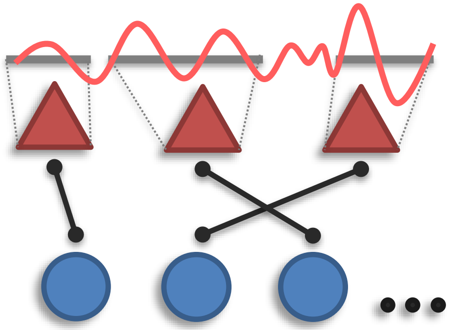
Soft nearest neighbor: $$\widetilde{v} = \sum_j^M \alpha_j v_j,$$

Find the nearest neighbor the other way and then penalize the distance:

$$\beta_k = \frac{e^{-||\widetilde{v} - u_k||^2}}{\sum_j^N e^{-||\widetilde{v} - u_j||^2}}$$

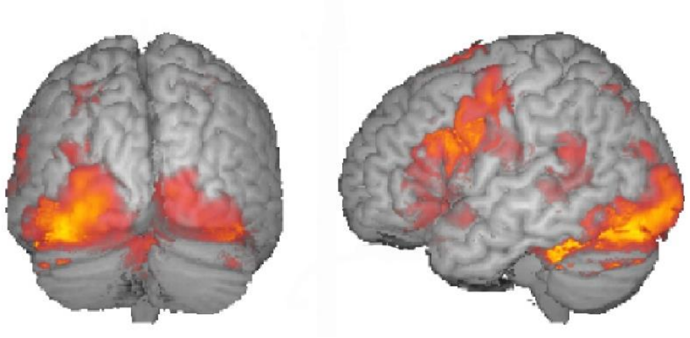
$$L_{cbr} = \frac{|i - \mu|^2}{\sigma^2} + \lambda \log(\sigma)$$

Discretization (aka Segmentation)

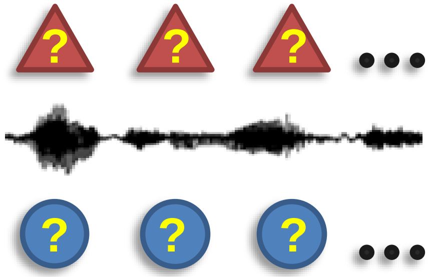


Common assumptions: ① Segmented elements

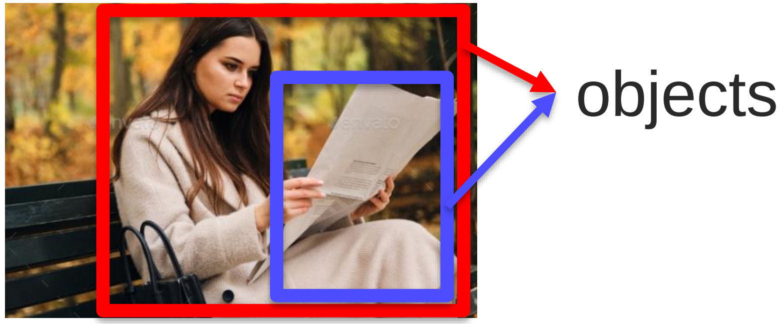
Examples:



Medical imaging



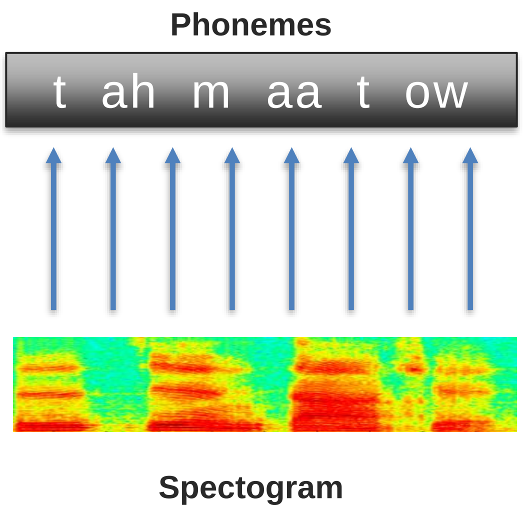
Signals



Images

Discretization – Example

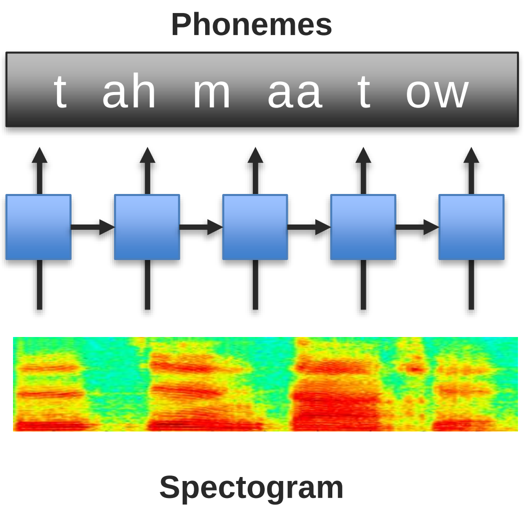
Sequence Labeling and Alignment



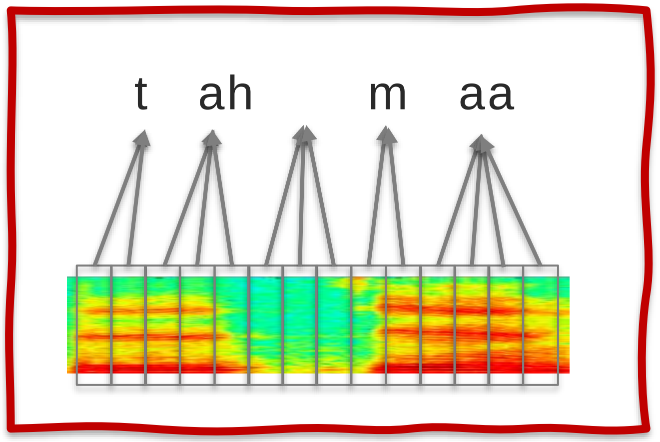
How can we predict the sequence of phoneme labels?

Discretization – Example

Sequence Labeling and Alignment



Challenge: many-to-1 alignment



How can we predict the sequence of phoneme labels?

Discretization – A Classification Approach

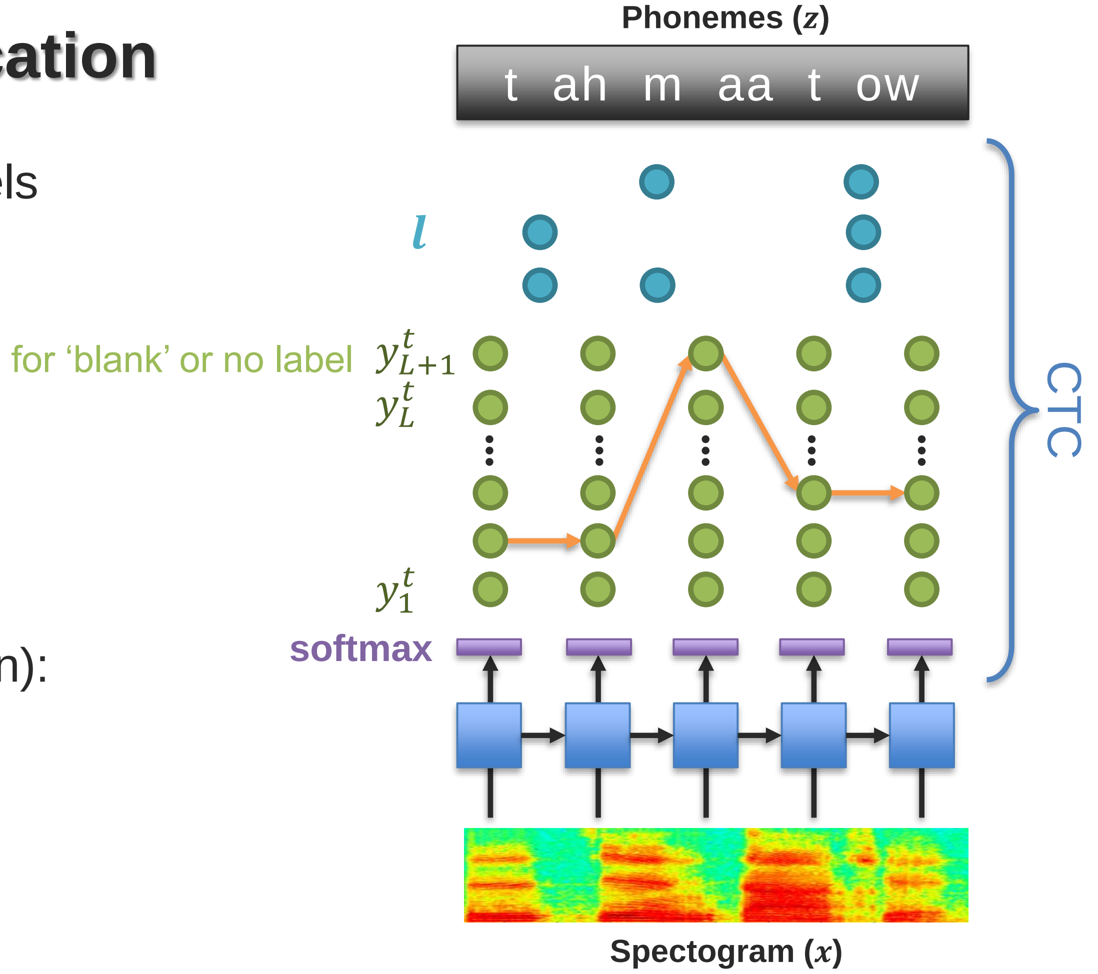
Connectionist Temporal Classification

4 Most probable sequence labels

3 Predicted labels l

2 Path π over the activations:

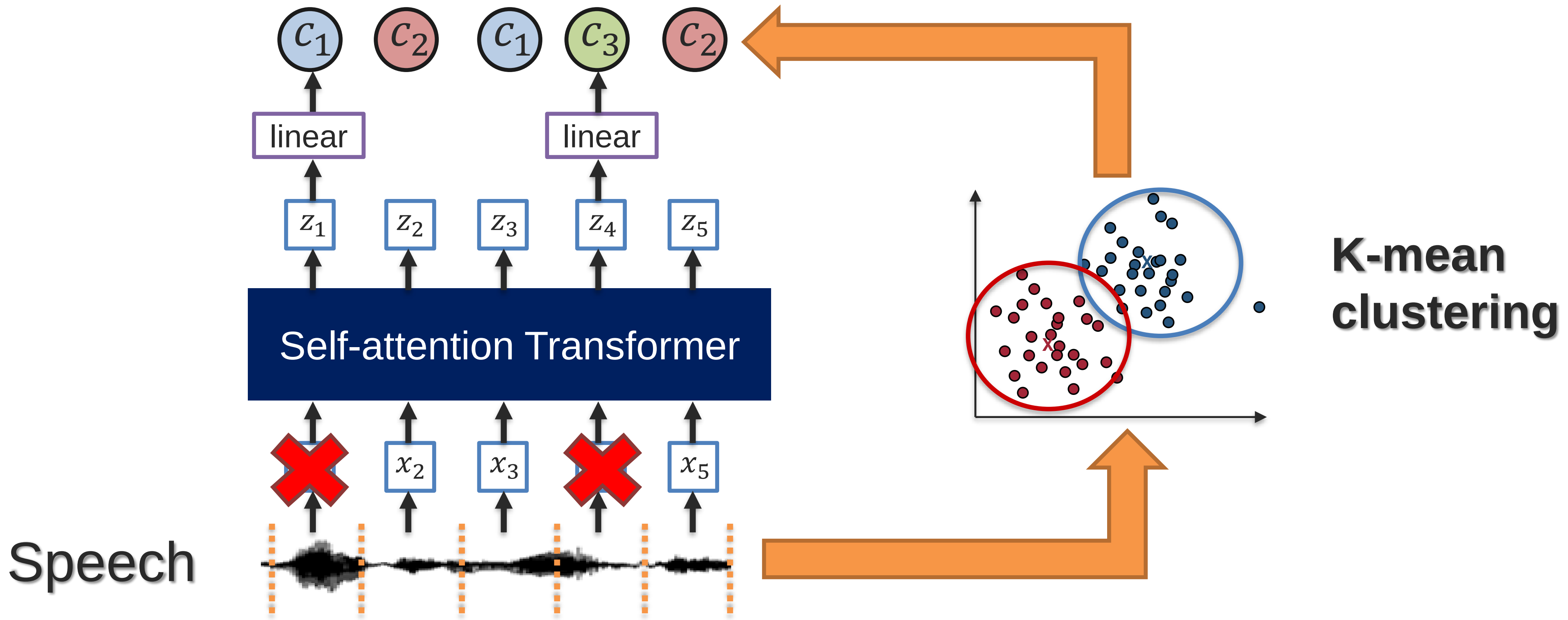
1 Output activations (distribution):



Grave et al., Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks, ICML 2006

Discretization and Representation – Cluster-based Approaches

HUBERT: Hidden-Unit BERT



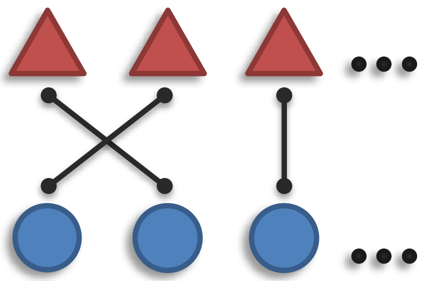
Hsu et al., HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units, arxiv 2021

Challenge 2: Alignment

Definition: Identifying and modeling cross-modal connections between all elements of multiple modalities, building from the data structure

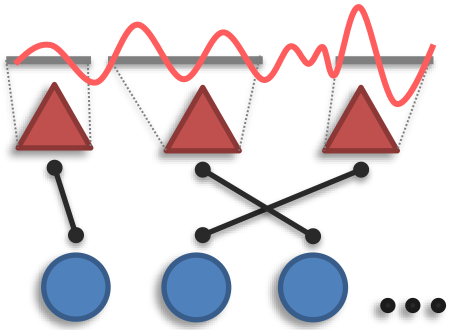
Sub-challenges:

Discrete Alignment



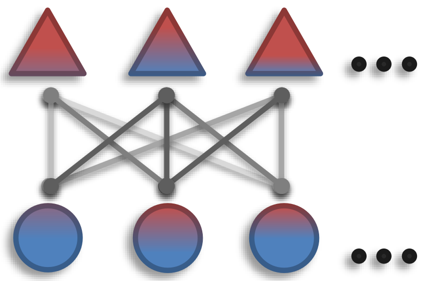
Discrete elements and connections

Continuous Alignment



Segmentation and continuous warping

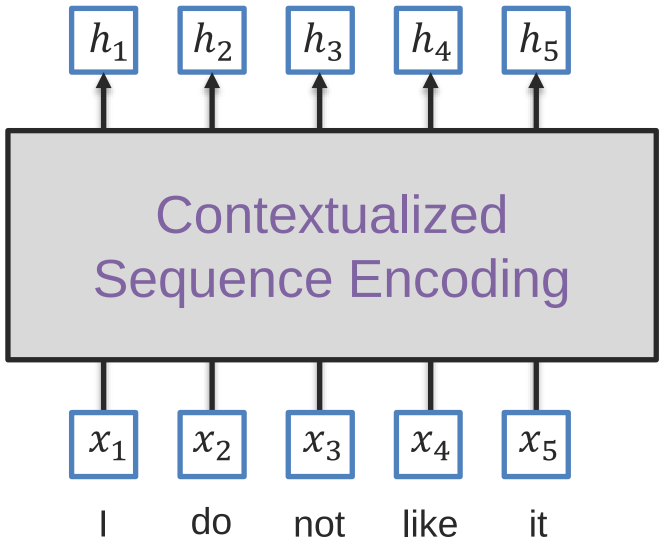
Contextualized Representation



Alignment
+ representation

Contextualized Sequence Representations

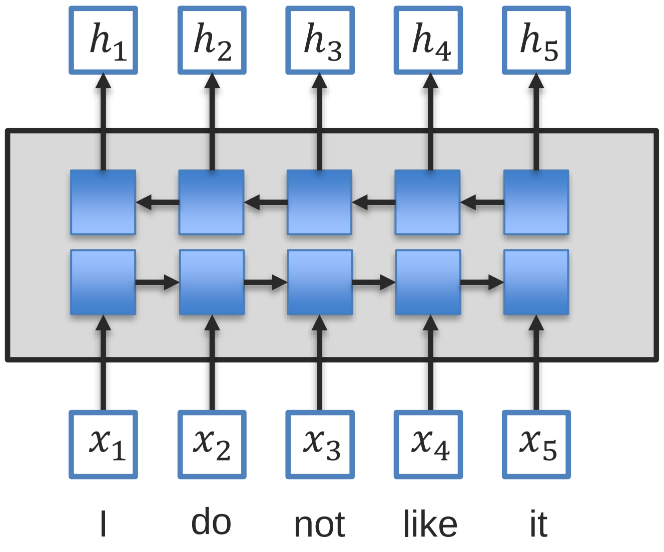
Sequence Encoding - Contextualization



How to encode this sequence while modeling the interaction between elements (e.g., words)?

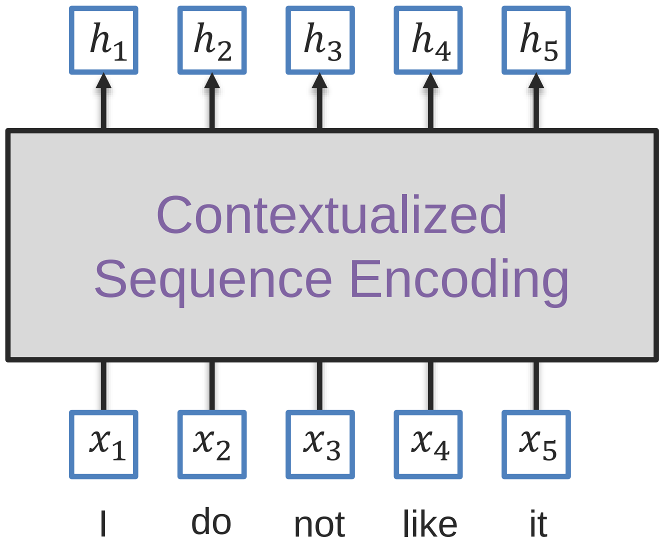
Option 1: Bi-directional LSTM:

(e.g., ELMO)

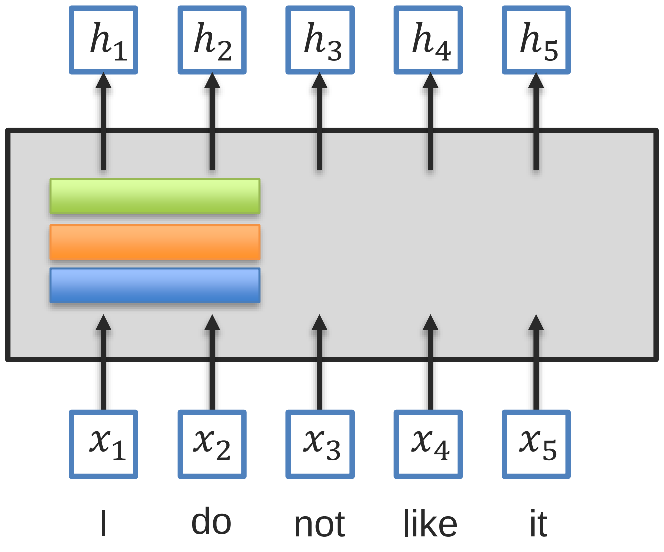


But harder to parallelize…

Sequence Encoding - Contextualization



Option 2: Convolutions

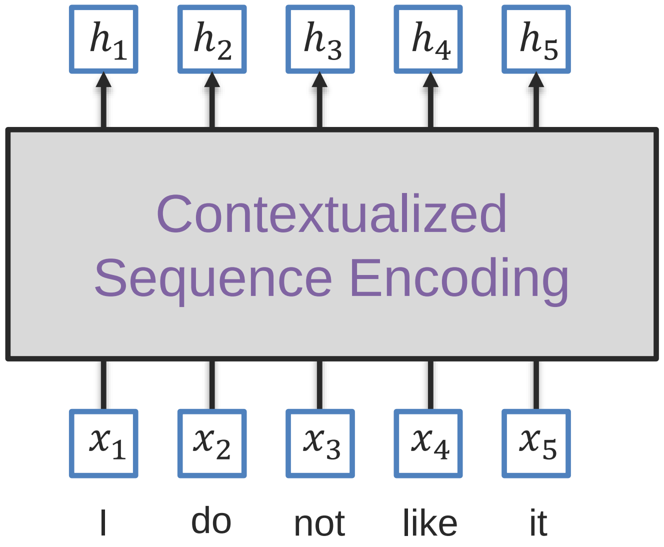


Can be parallelized!

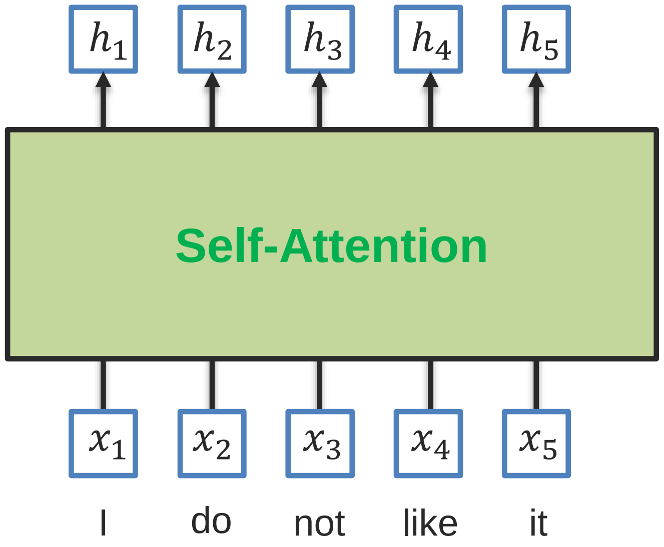
But modeling long-range dependencies require multiple layers

And convolutional kernels are static

Sequence Encoding - Contextualization



Option 3: Self-attention



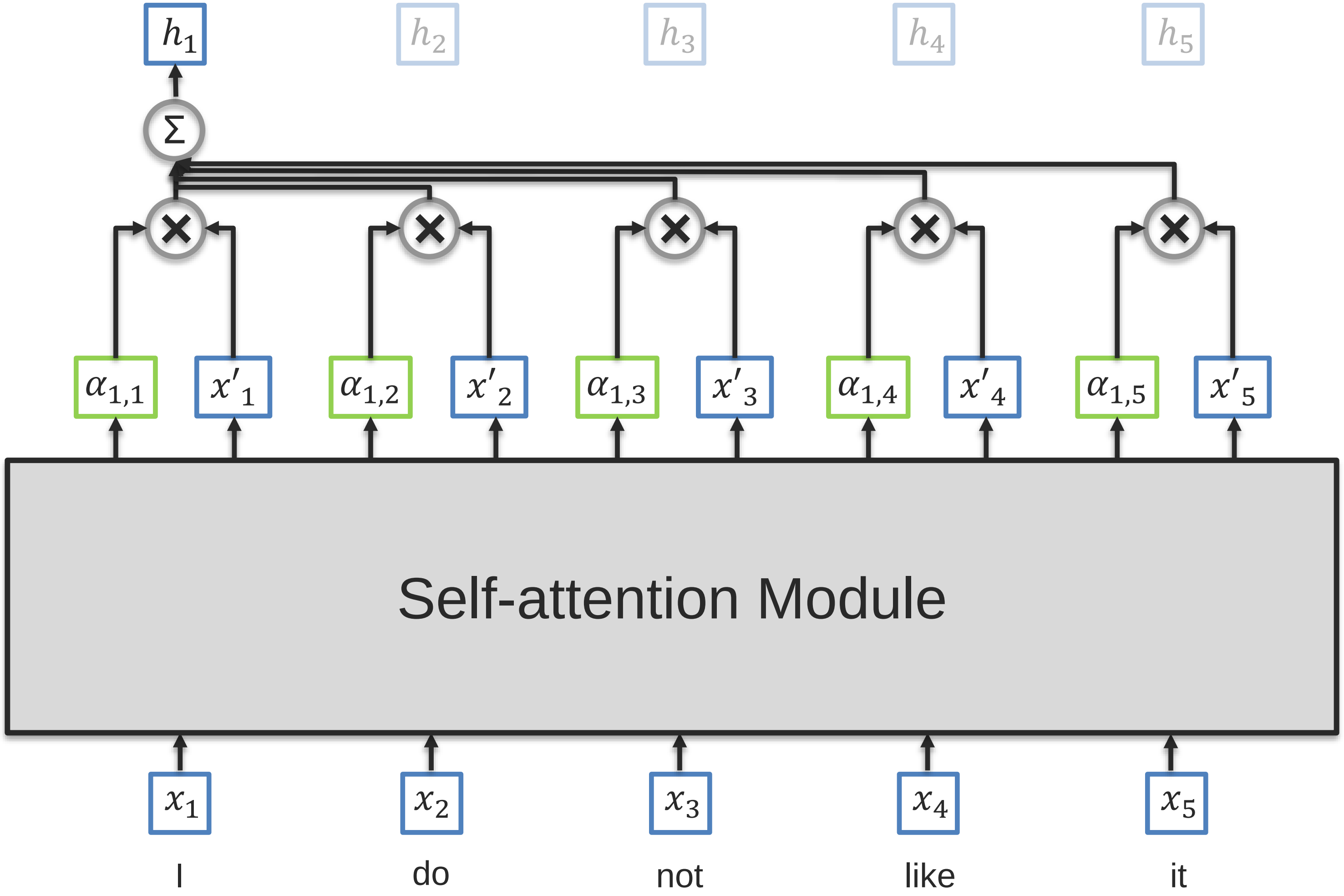
Can be parallelized!

Long-range dependencies

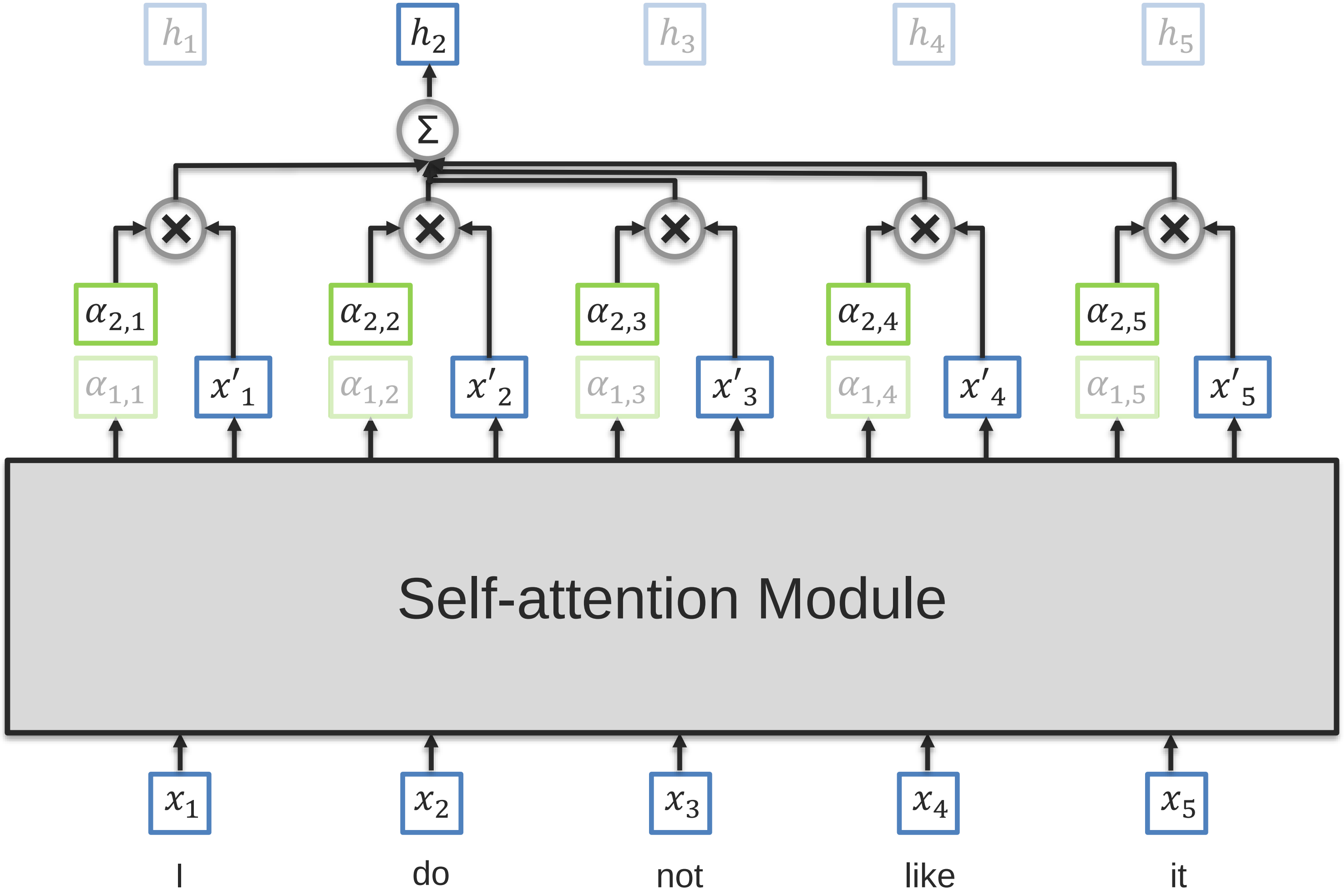
Dynamic attention weights

Self-Attention

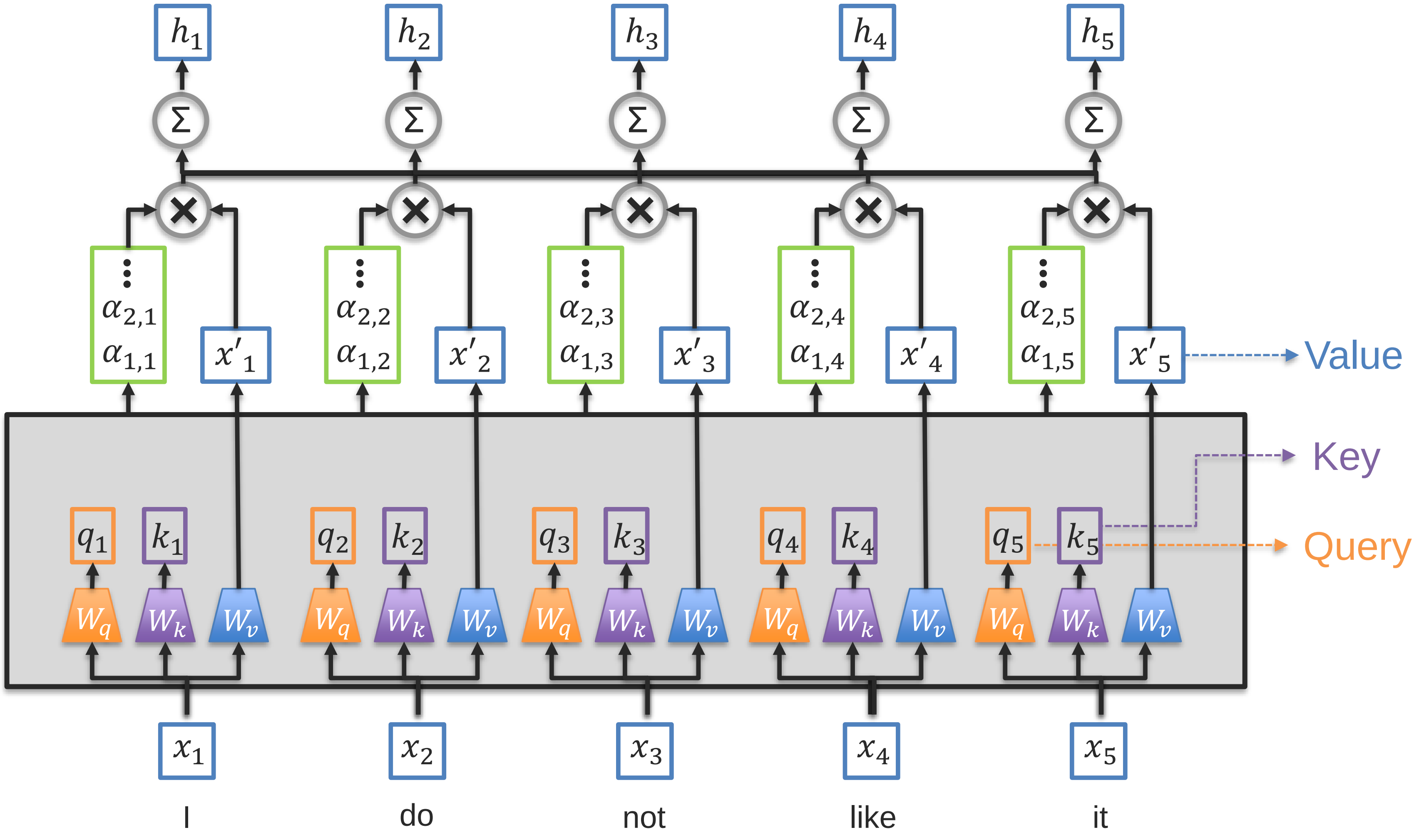
Self-Attention



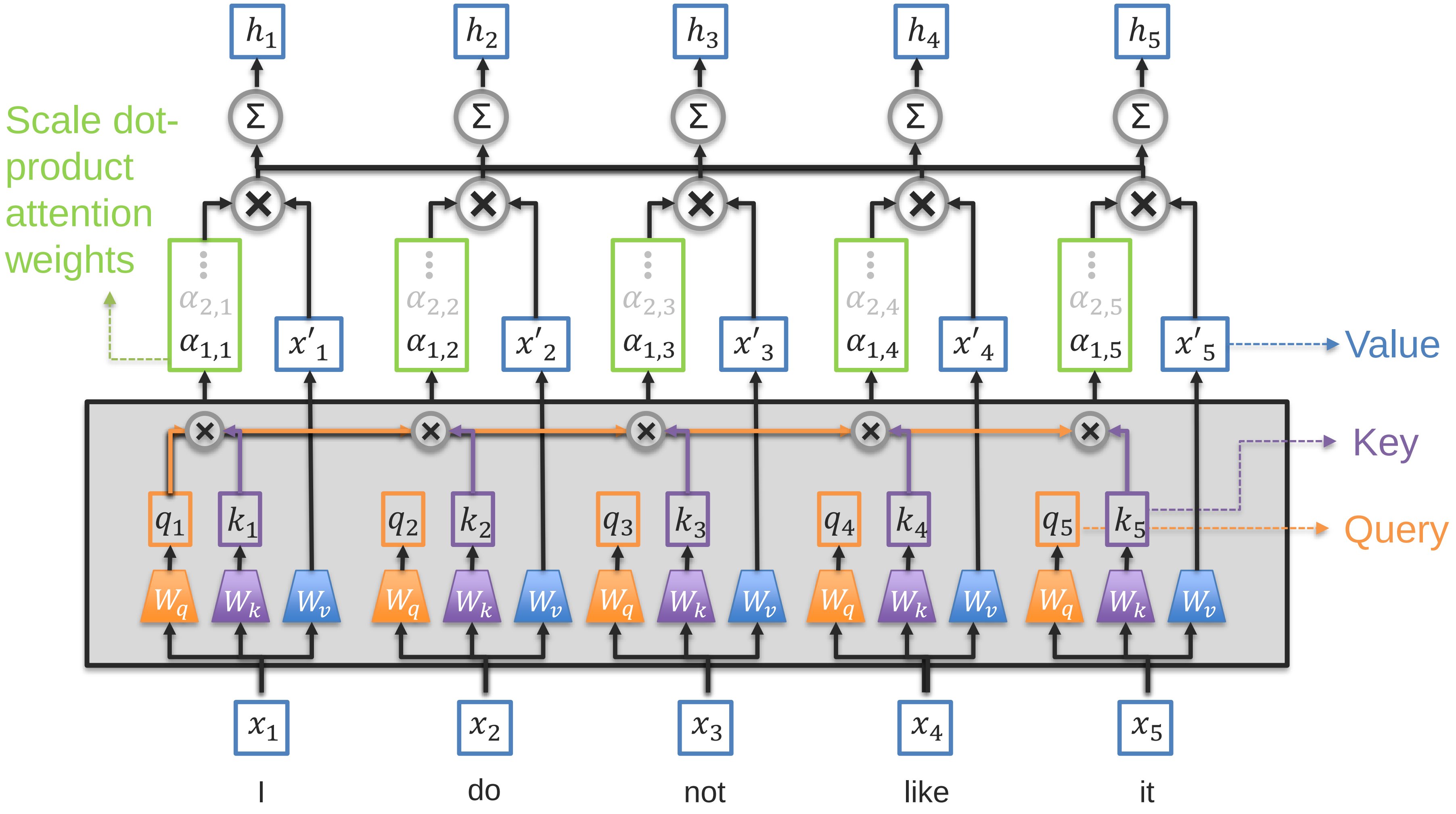
Self-Attention



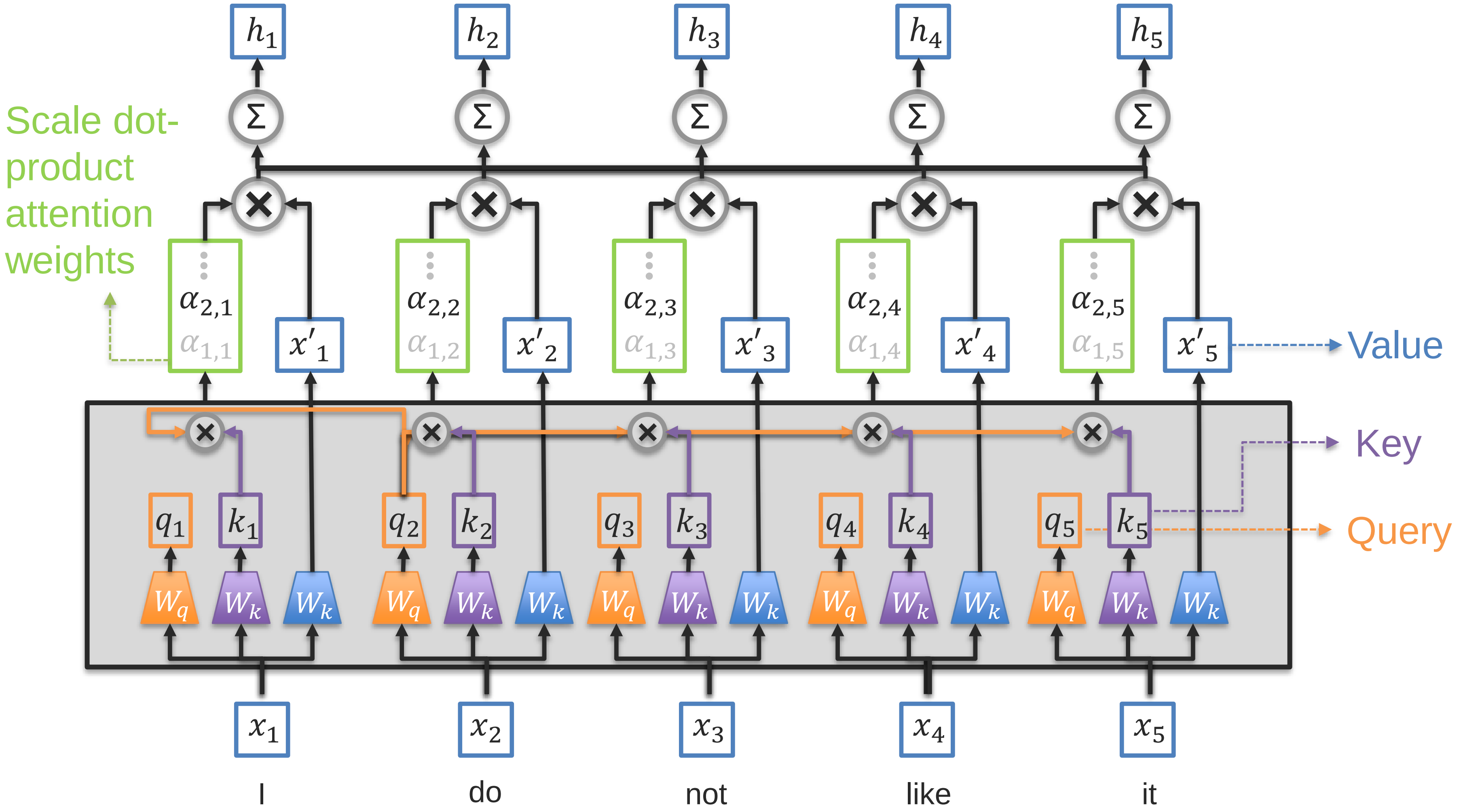
Transformer Self-Attention



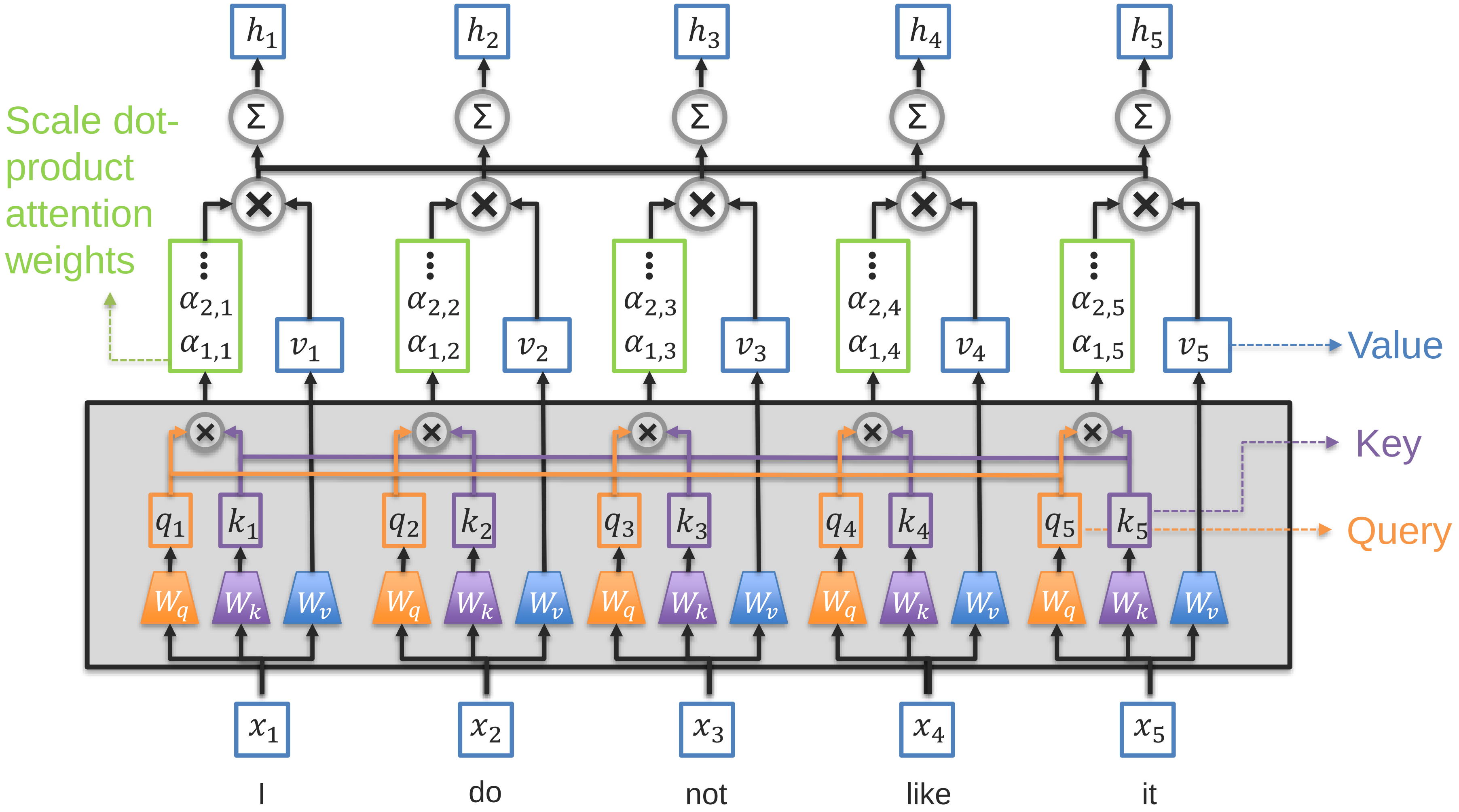
Transformer Self-Attention



Transformer Self-Attention

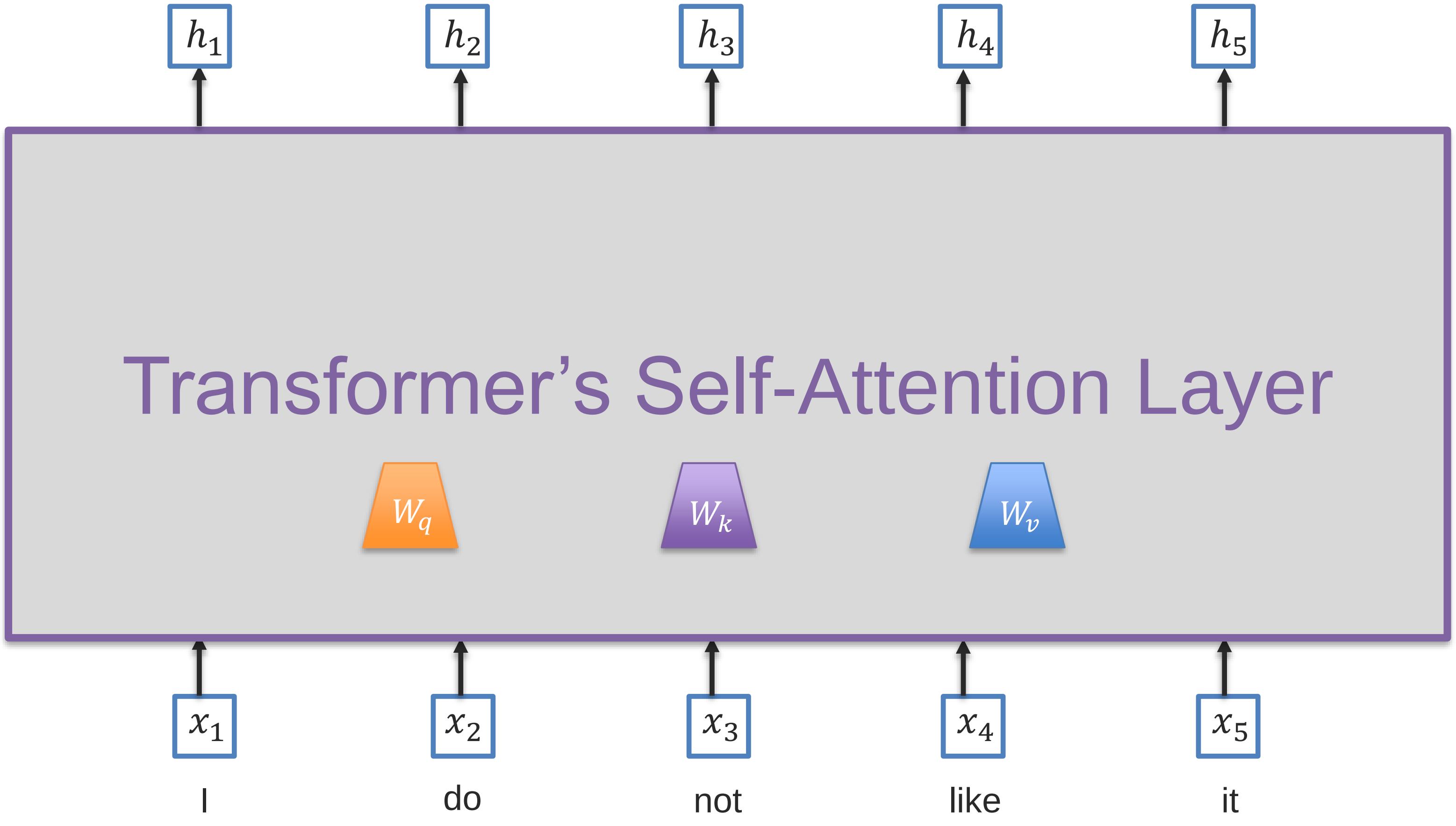


Transformer Self-Attention

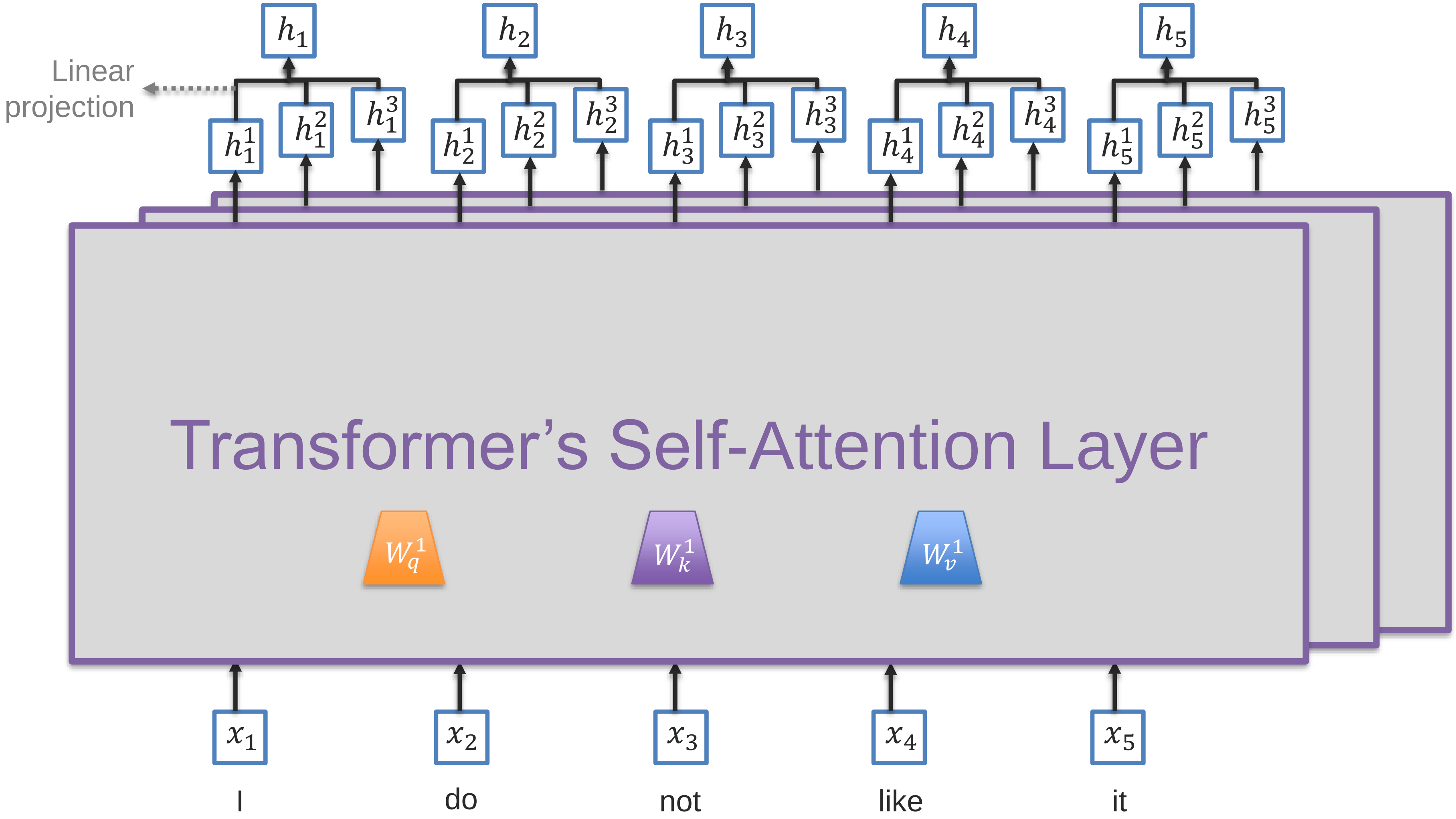


Transformer Self-Attention

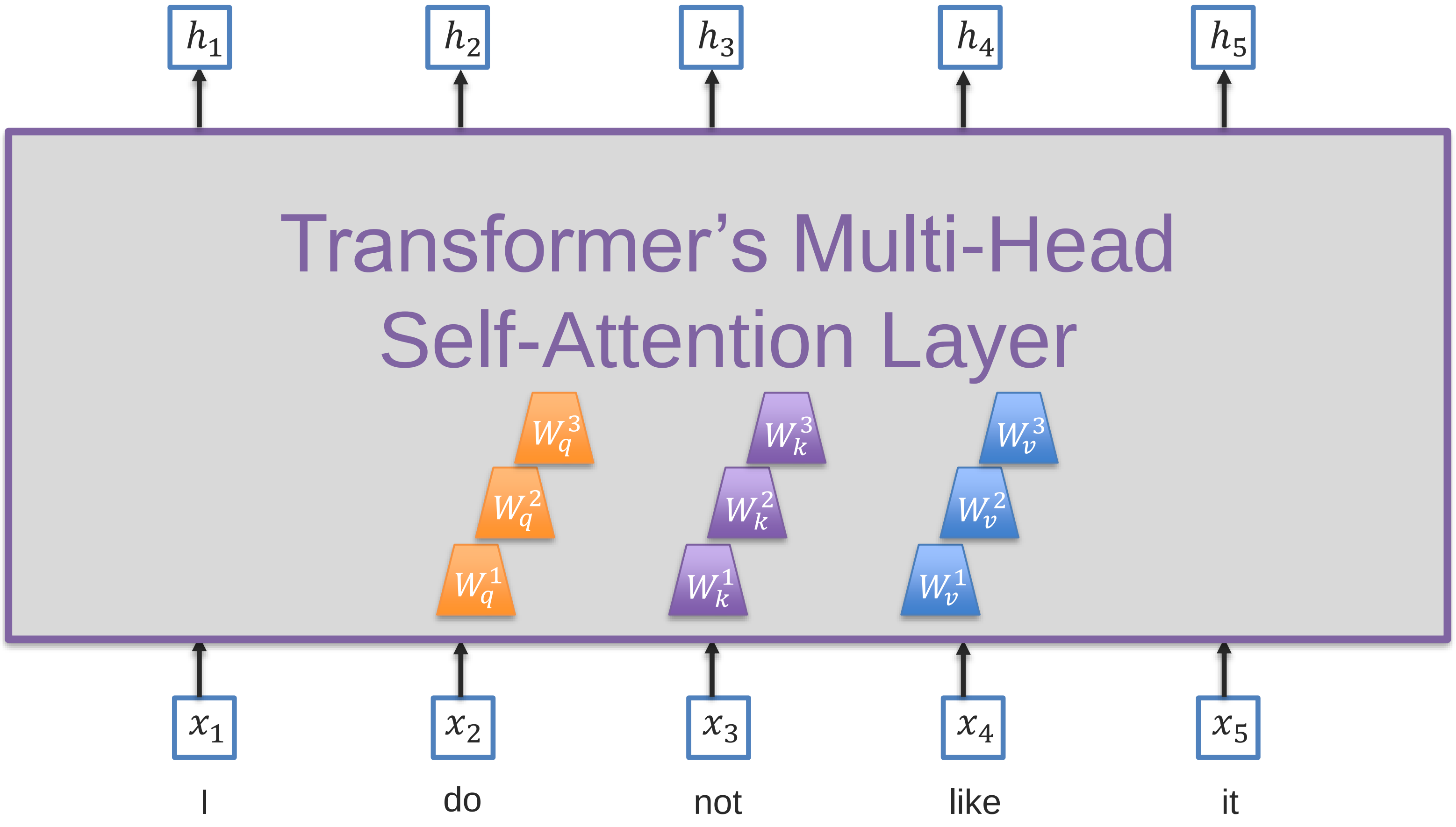
What if we want to attend simultaneously to multiple subspaces of $\boldsymbol{x}$?



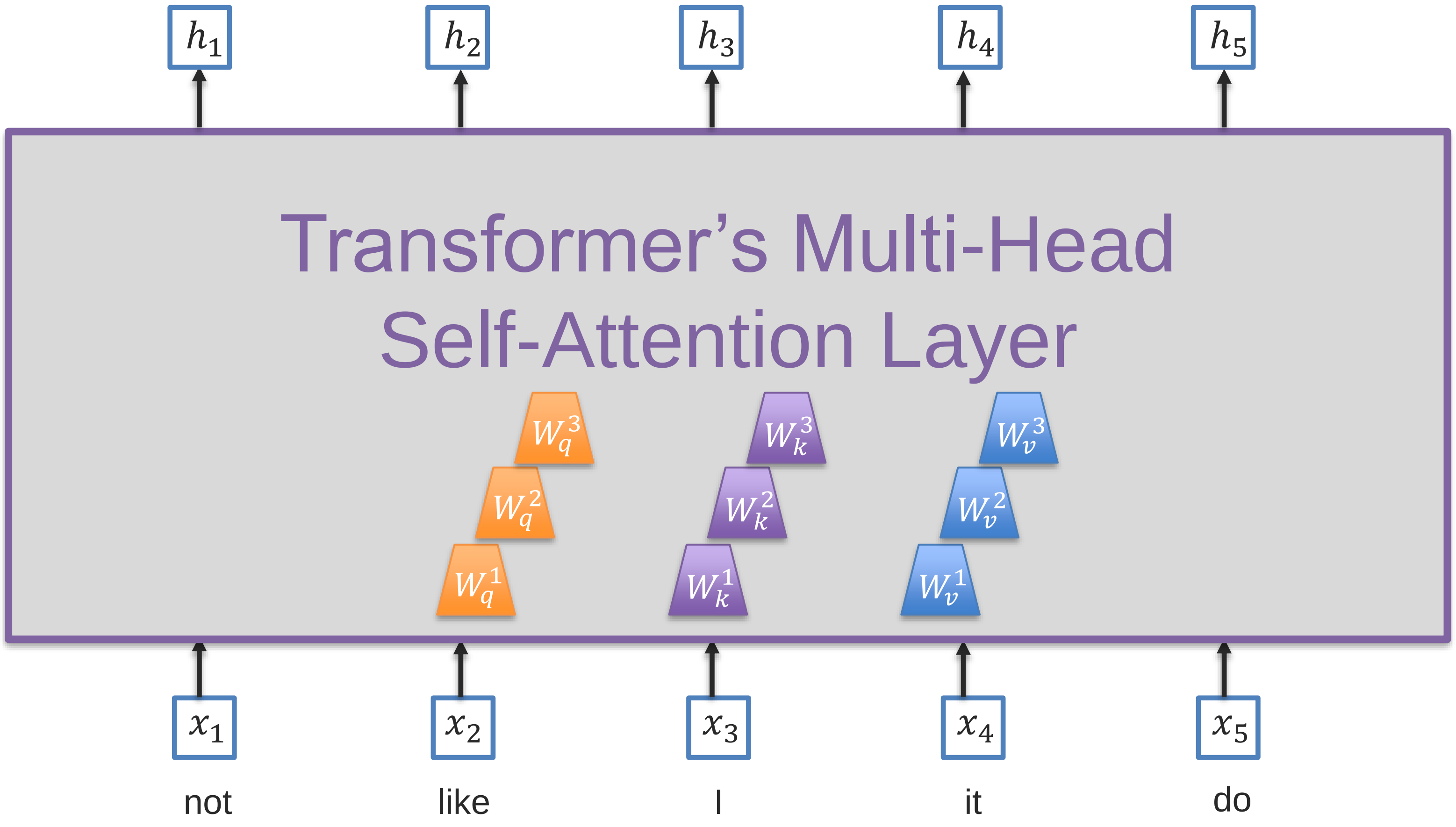
Transformer Multi-Head Self-Attention



Transformer Multi-Head Self-Attention



Transformer Multi-Head Self-Attention



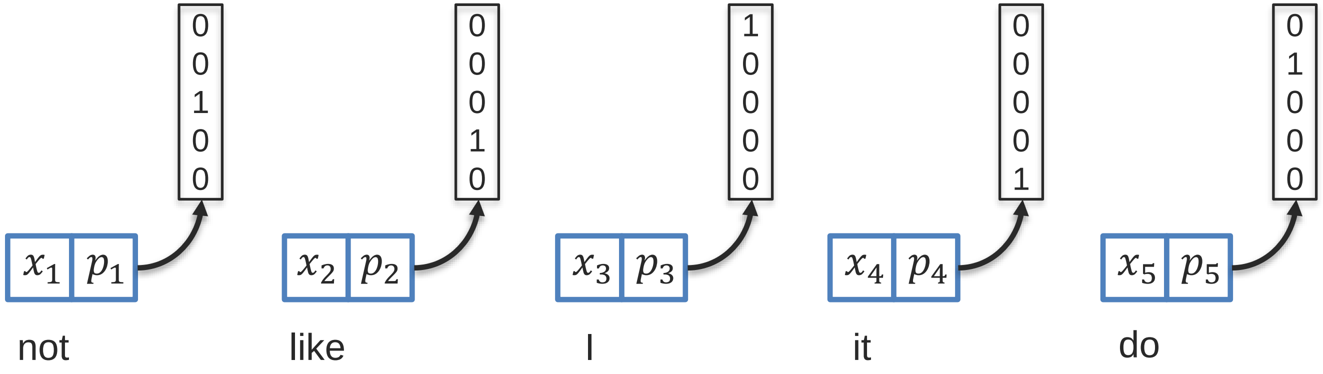
What happens if the words are shuffled?

Position embeddings

- Position information is not encoded in a self-attention module

How can we encode position information?

Simple approach: one-hot encoding

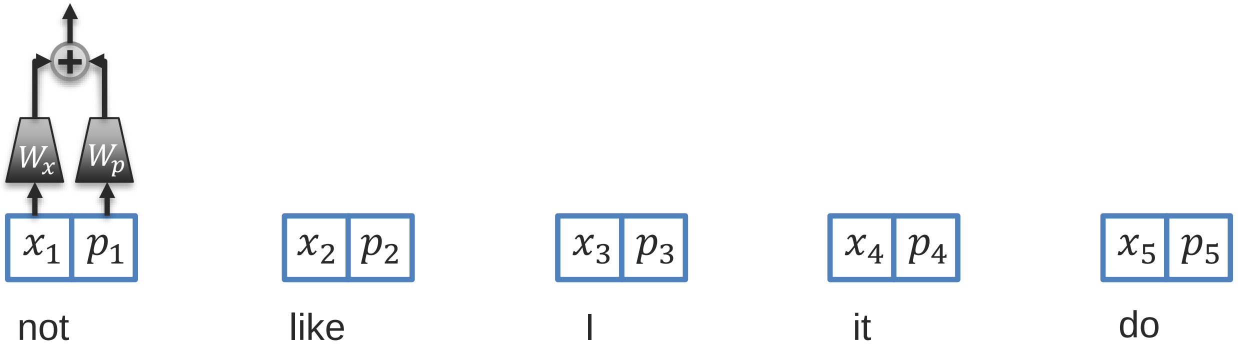


Position embeddings

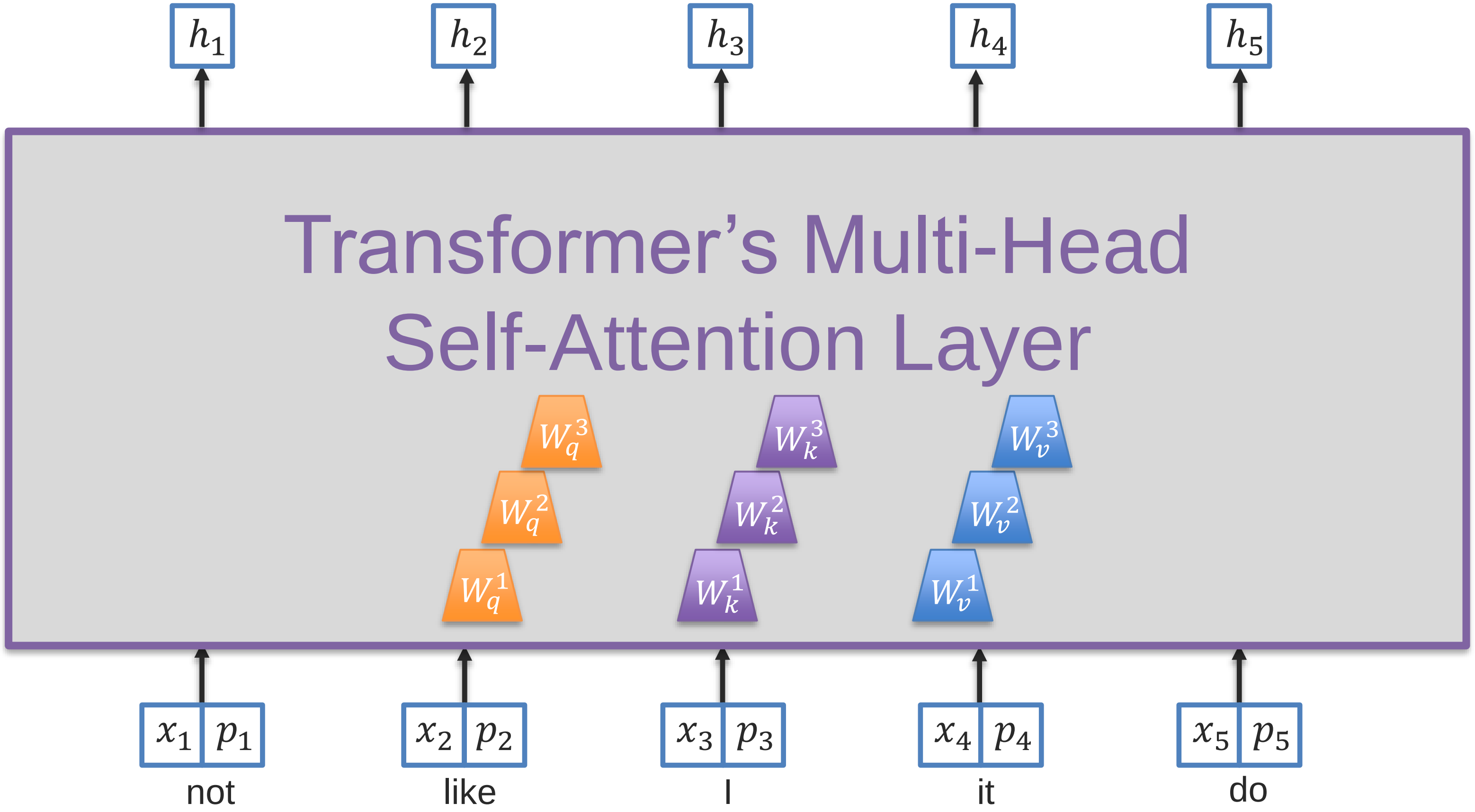
- ❑ Position information is not encoded in a self-attention module

How can we encode position information?

Simple approach: one-hot encoding + linear embeddings + { Sum - or - concat }

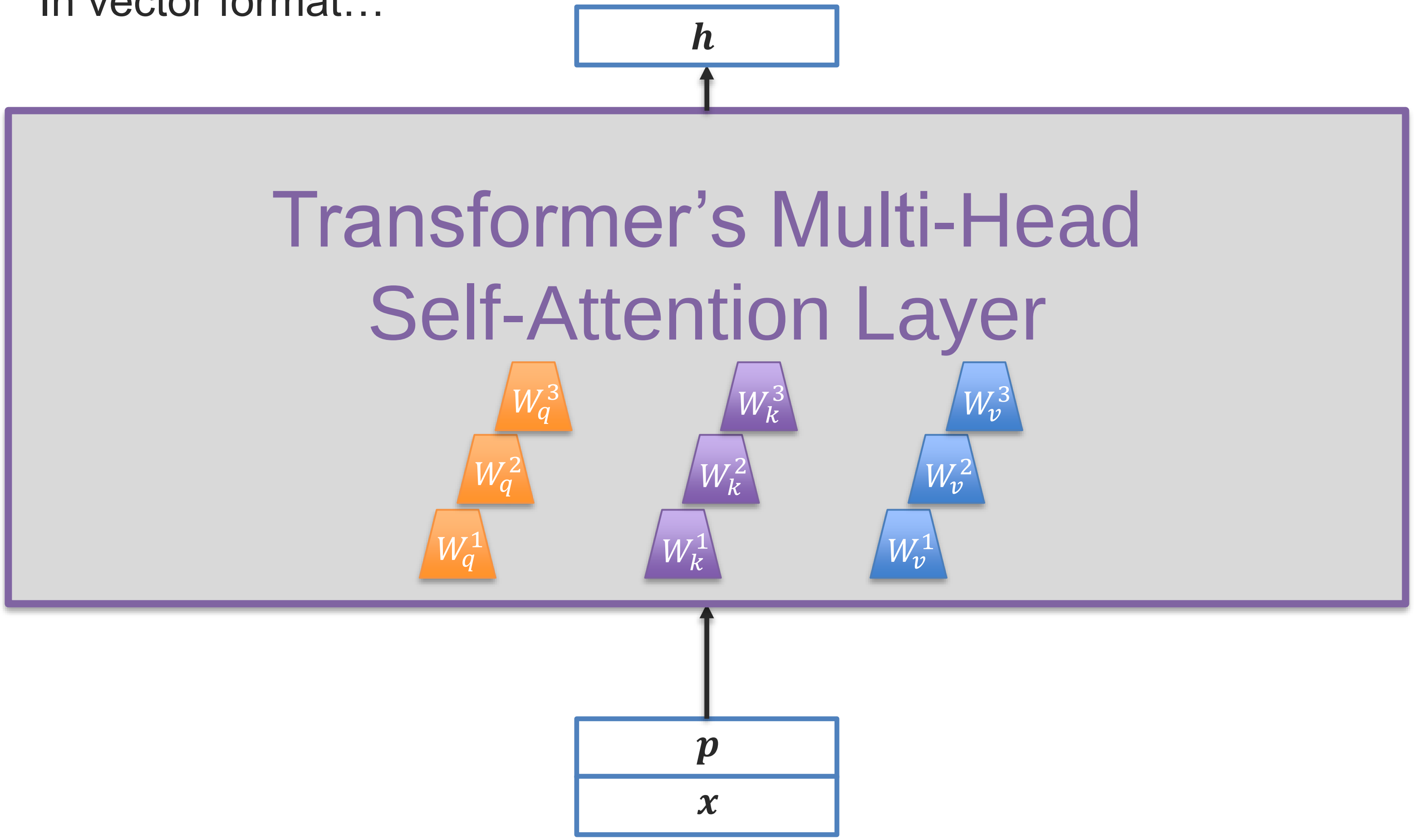


Transformer Multi-Head Self-Attention

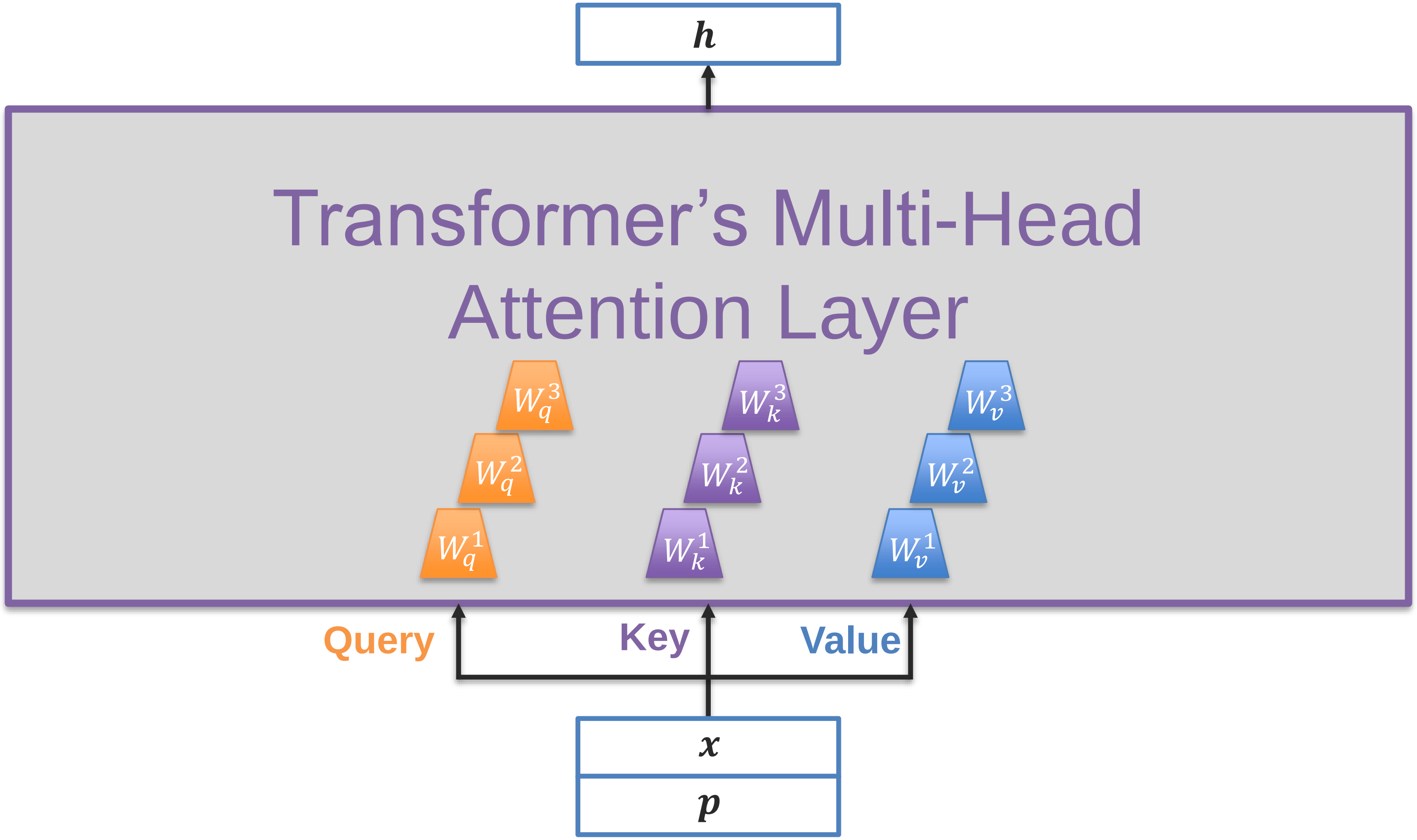


Transformer Multi-Head Self-Attention

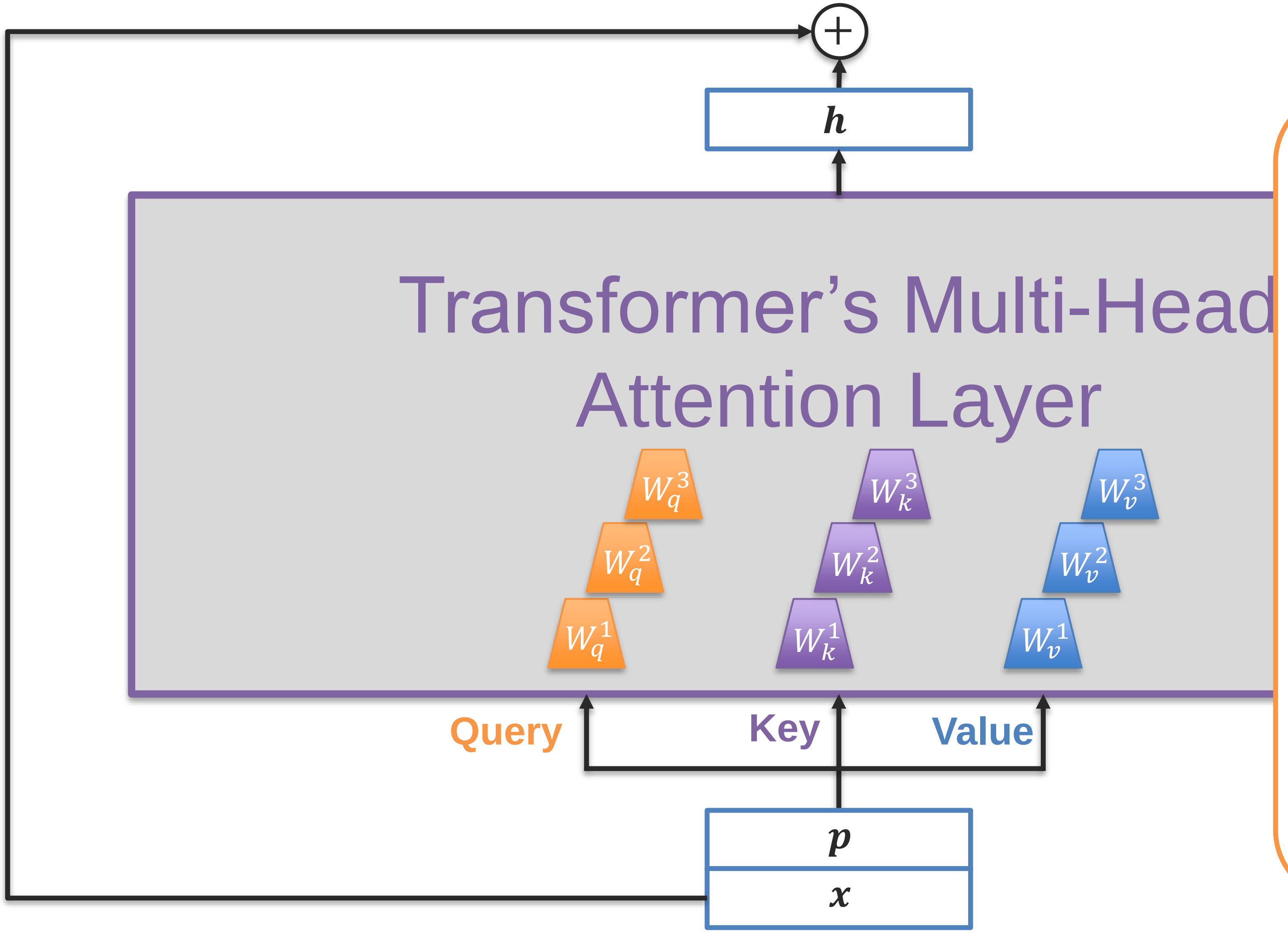
In vector format…

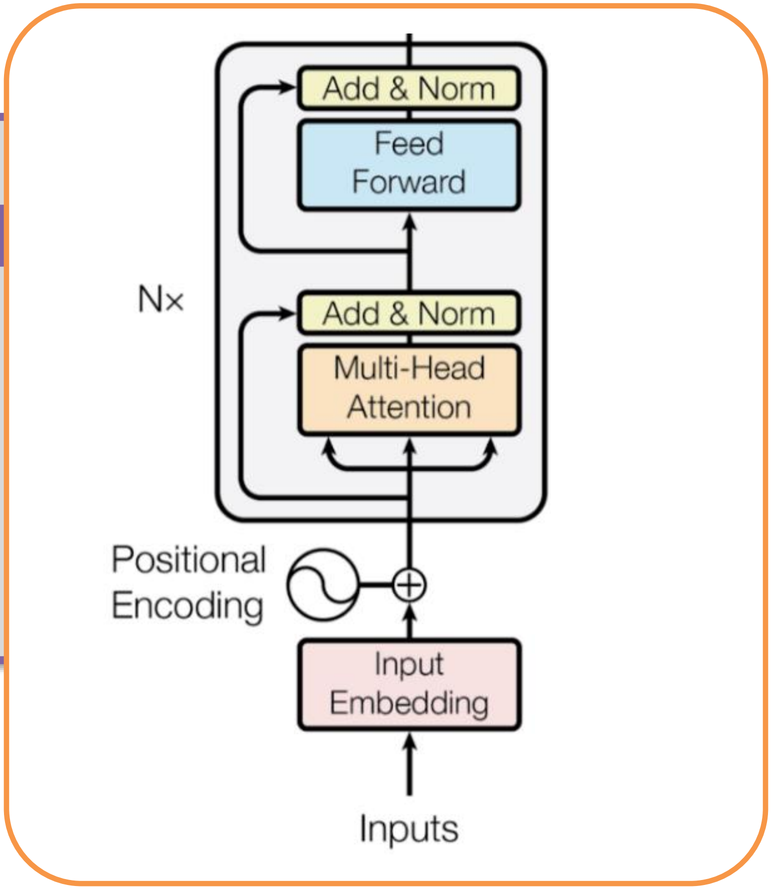


Transformer Multi-Head Attention



Transformer – Residual Connection

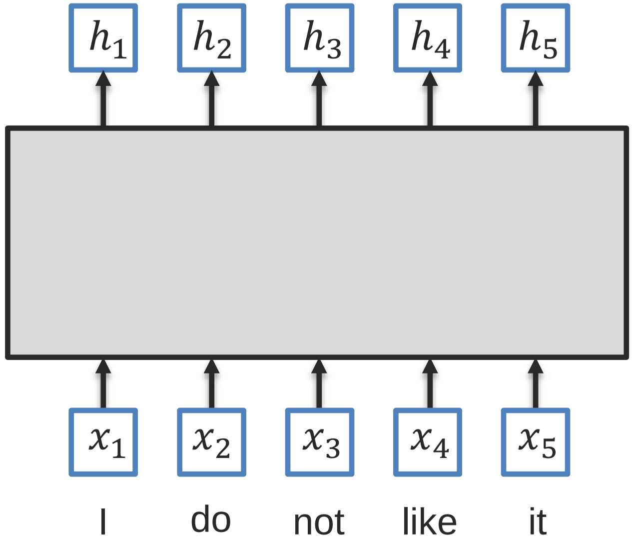




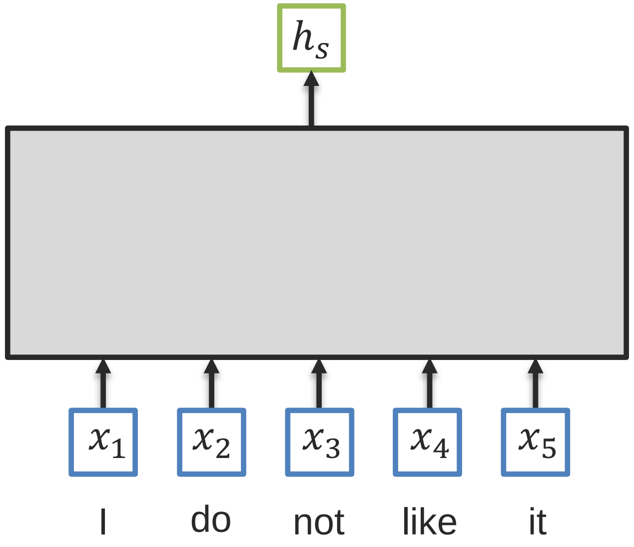
Language Pre-training

Token-level and Sentence-level Embeddings

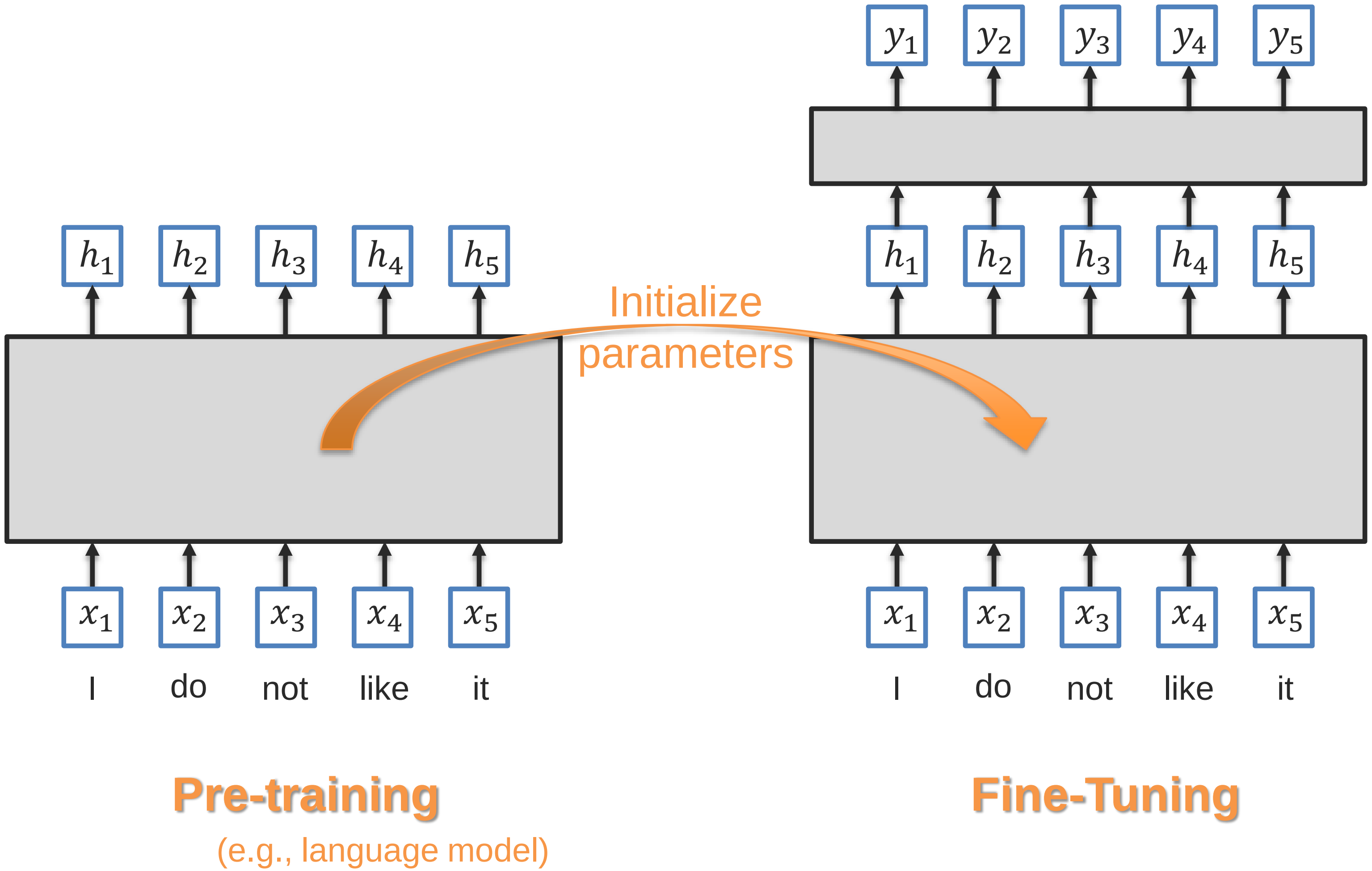
Token-level embeddings



Sentence-level embedding



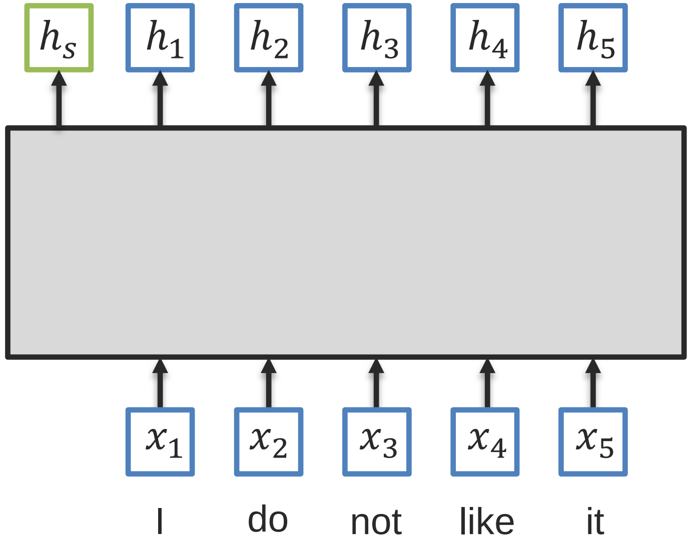
Pre-Training and Fine-Tuning



BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning

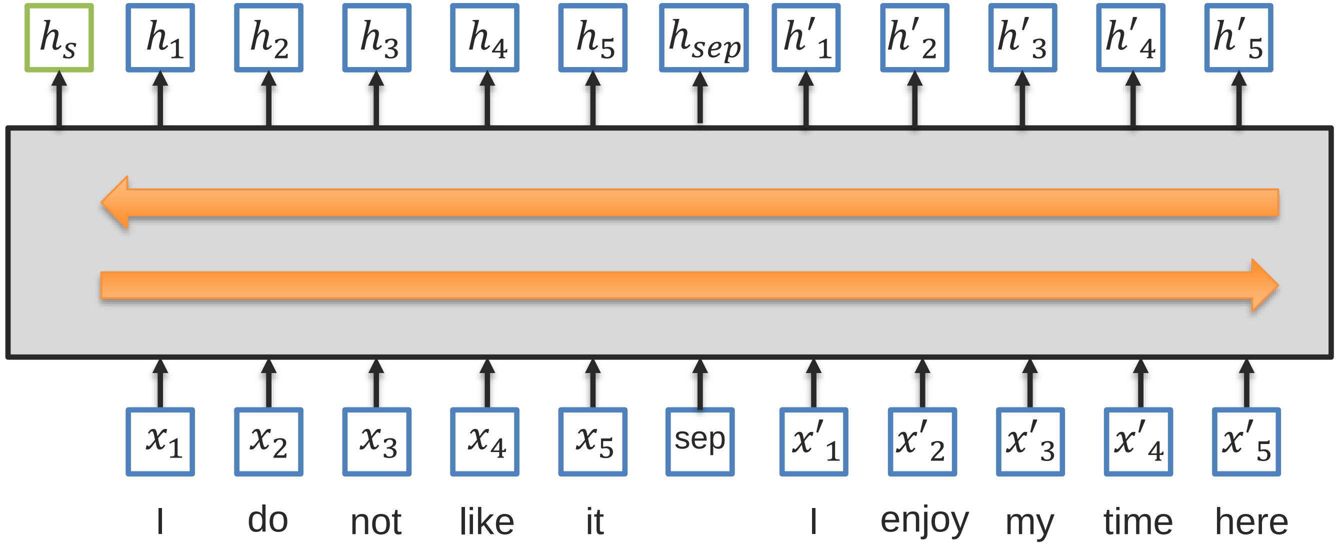


BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning
3. Can be used learn relationship between sentences
4. Models bidirectional and long-range interactions between tokens

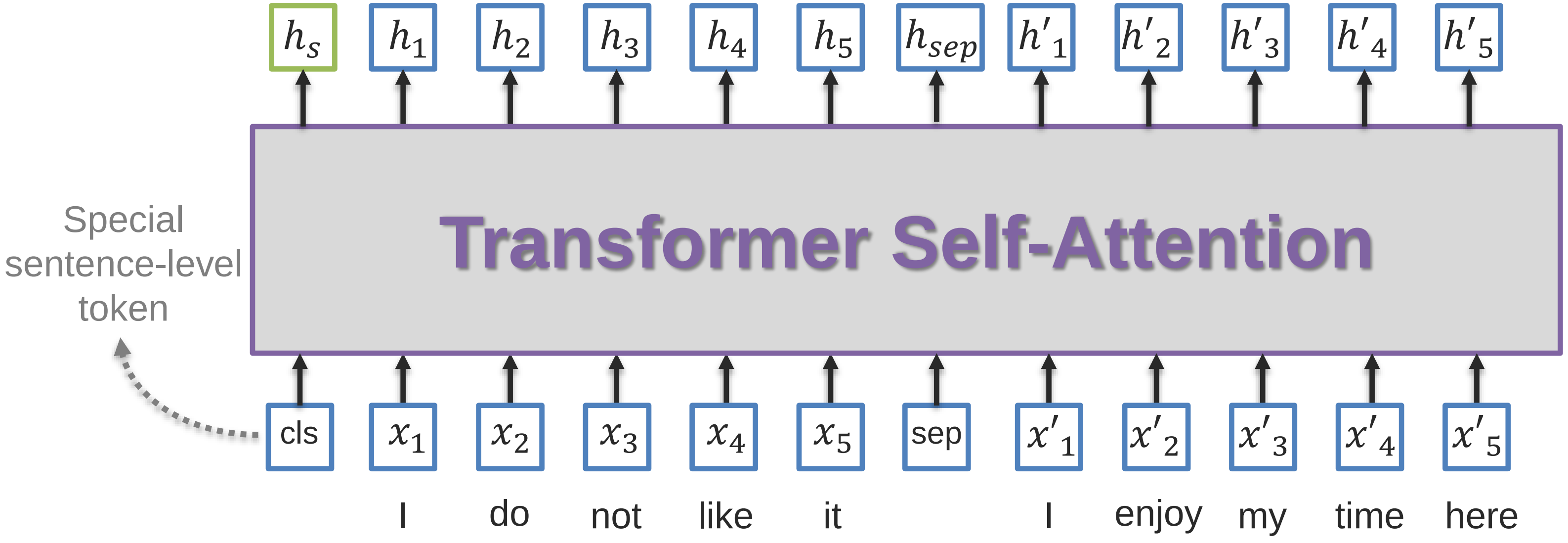
How can we do all this?



BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning
3. Can be used learn relationship between sentences
4. Models bidirectional interactions between tokens



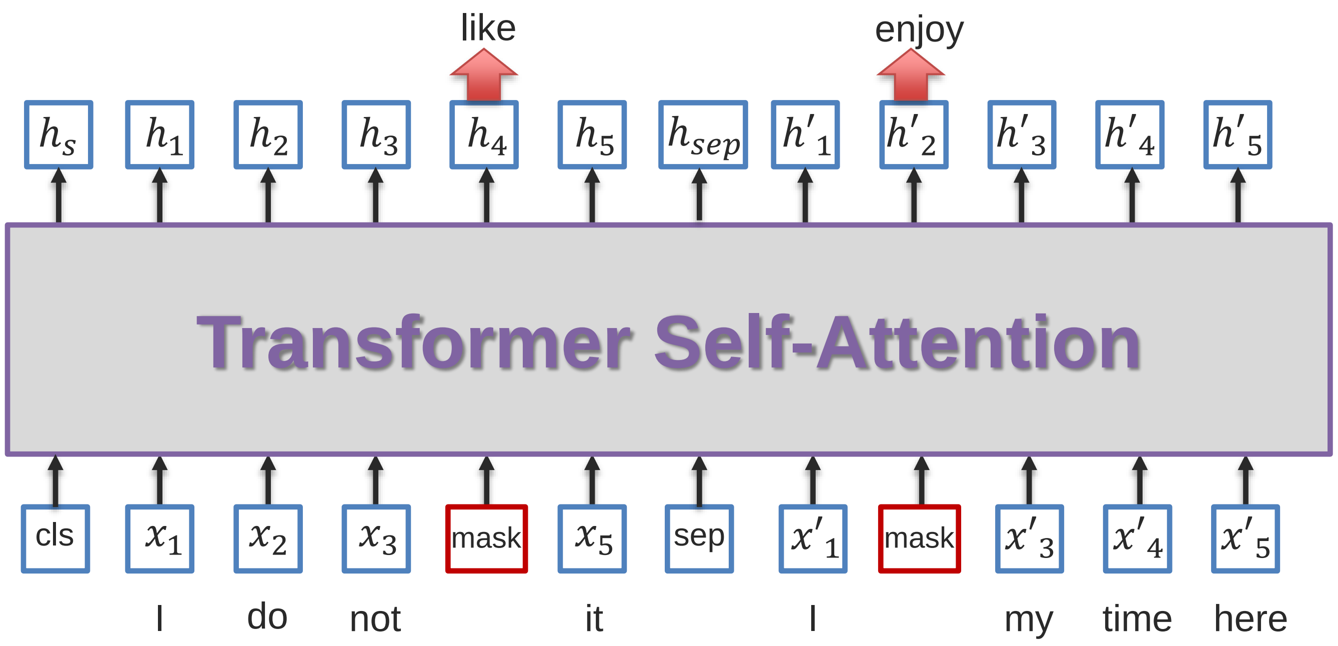
But how to train self-supervised?

Pre-training BERT Model

1 **Masked Language Model**

Randomly mask input tokens and then try to predict them

What is the loss function?



Pre-training BERT Model

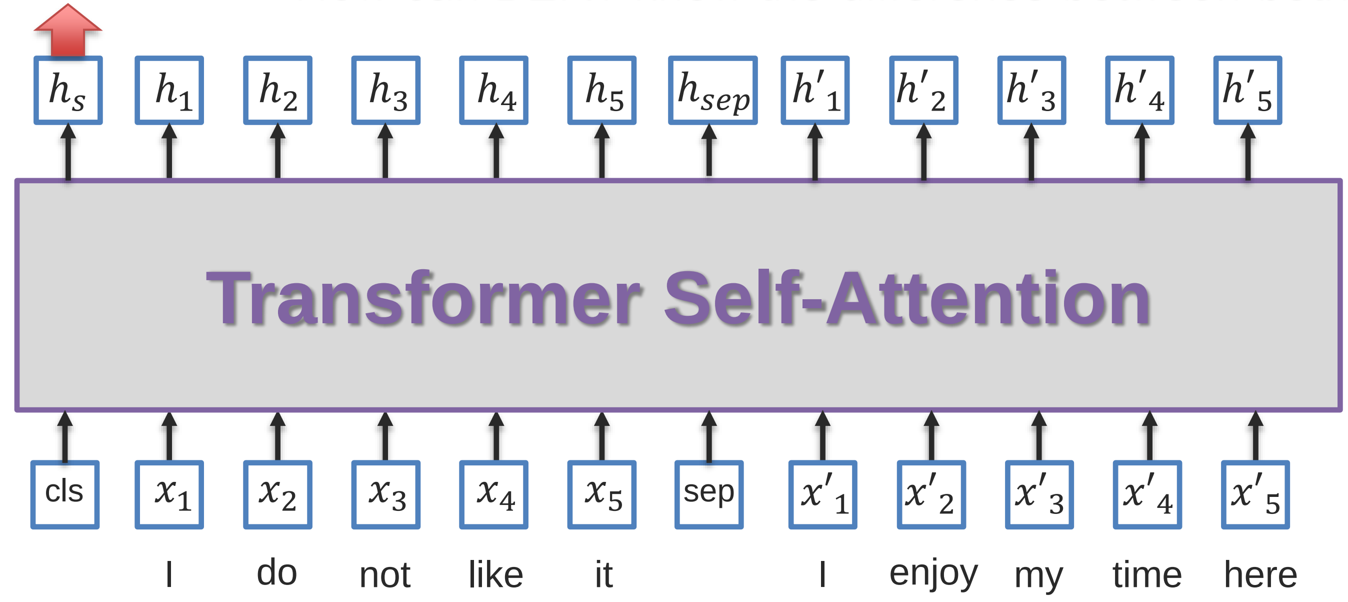
2 **Next Sentence Prediction**

Given two sentences, predict if this is the next one or not

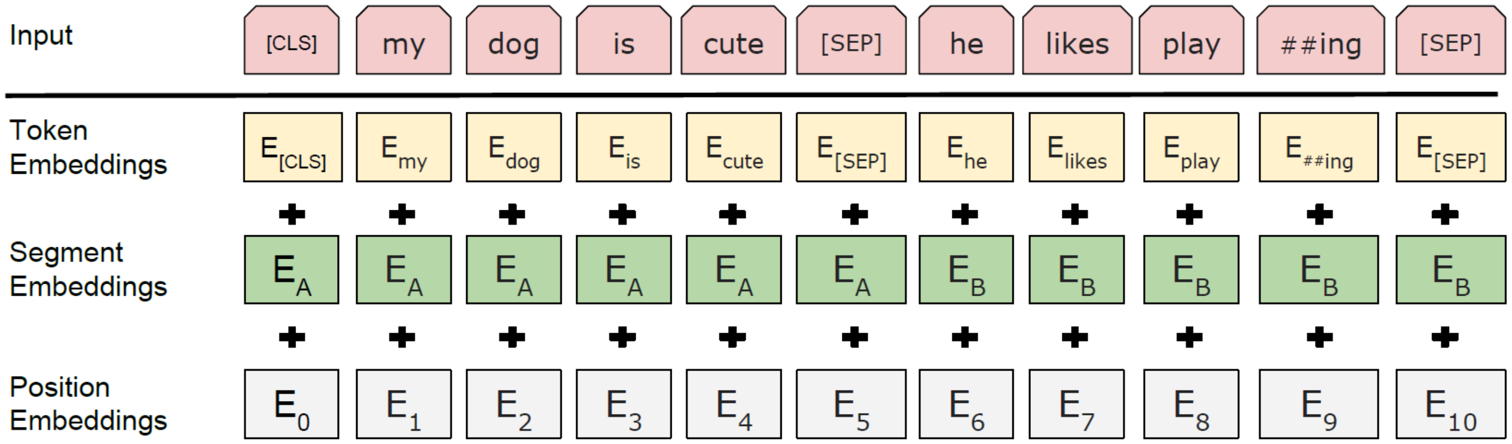
What is the loss function?

Where can we find training data?

How can BERT know the difference between both sentences?



Three Embeddings: Token + Position + Sentence

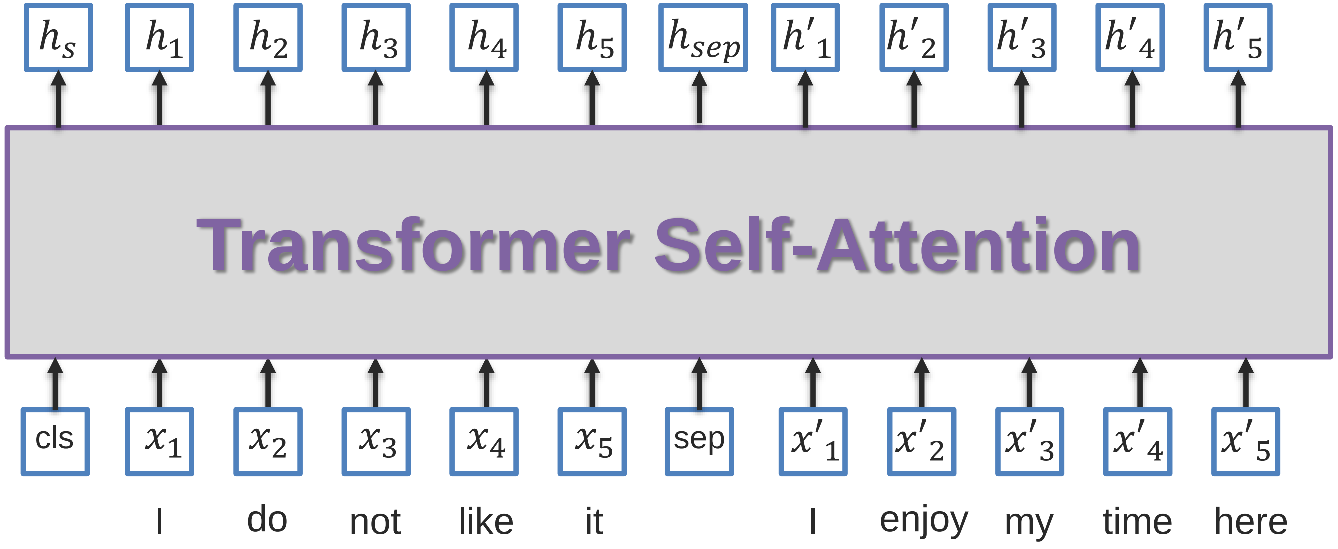


Fine-Tuning BERT

1 Sentence-level classification for only one sentence

Examples: sentiment analysis, document classification

How?

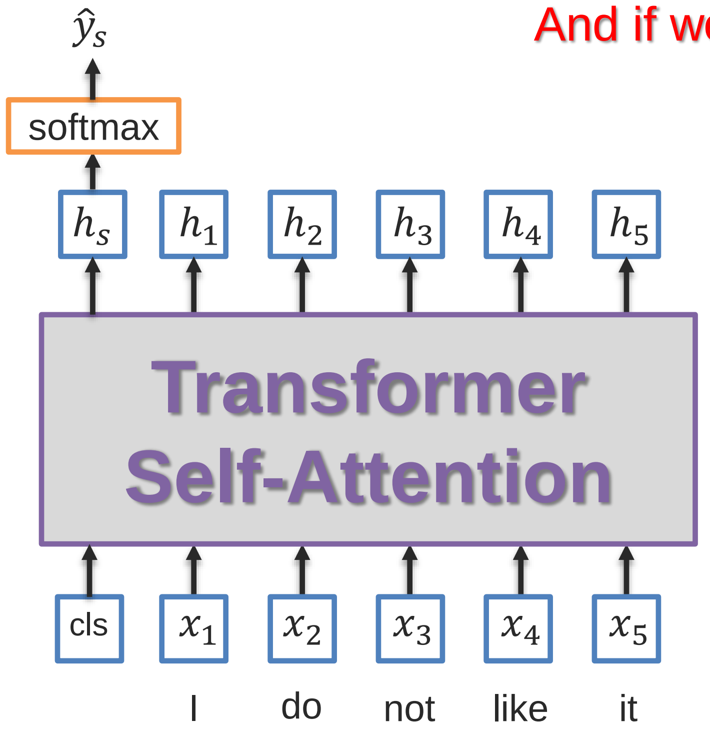


Fine-Tuning BERT

1 Sentence-level classification for only one sentence

Examples: sentiment analysis, document classification

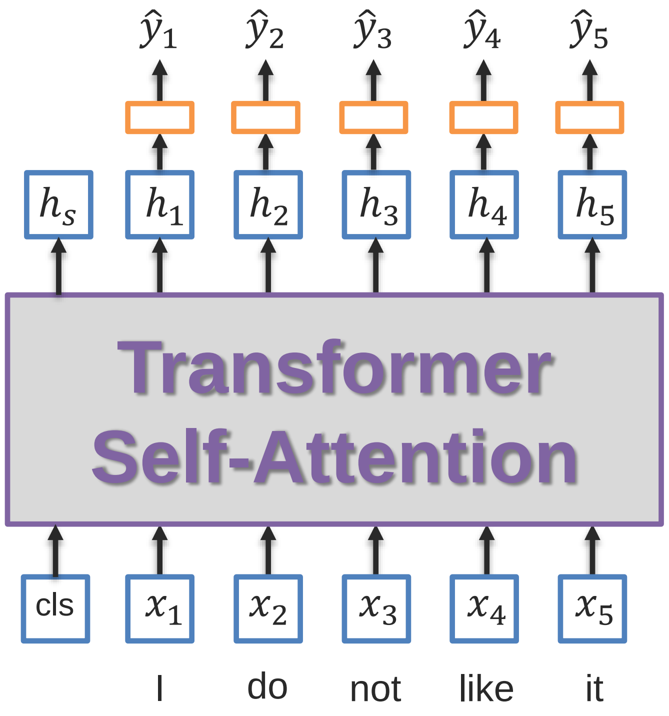
And if we have a label for each token?



Fine-Tuning BERT

2 Token-level classification for only one sentence

Examples: part-of-speech tagging, slot filling

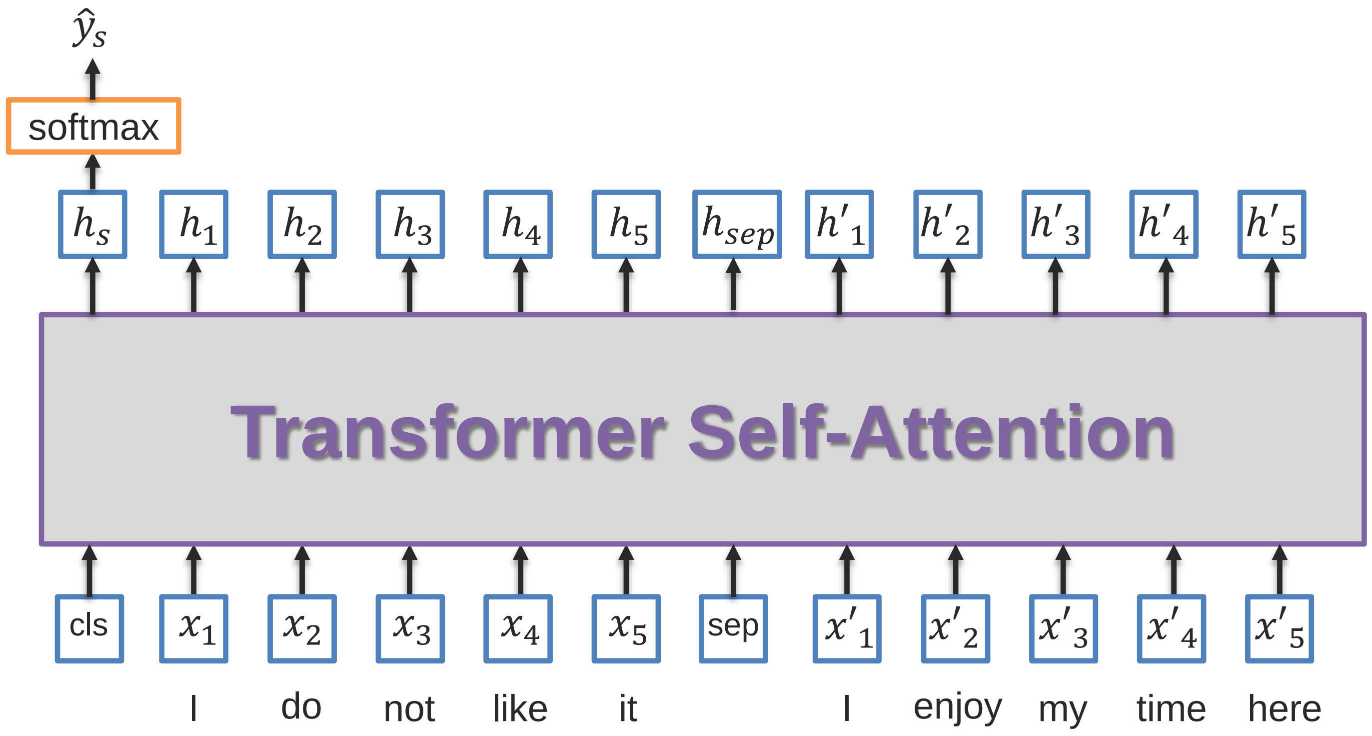


How to compare two sentences?

Fine-Tuning BERT

3 Sentence-level classification for two sentences

Examples: natural language inference



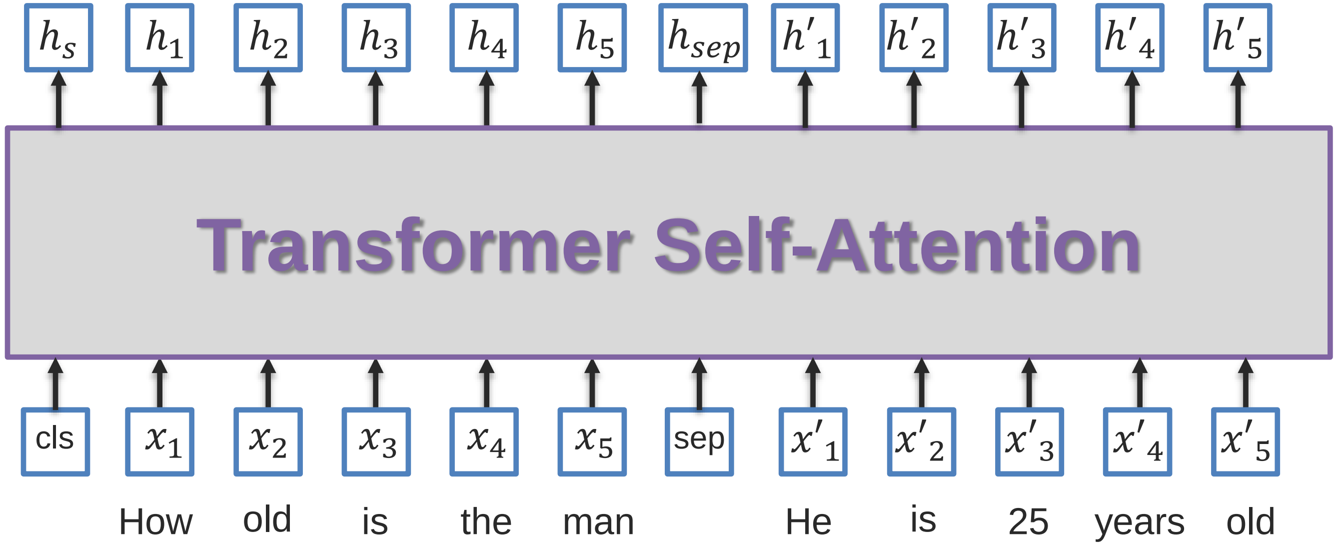
Fine-Tuning BERT

4 Question-answering: find start/end of the answer in the document

Paragraph: "*. . . Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.*"

Question 1: "*Which laws faced significant opposition?*"
Plausible Answer: *later laws*

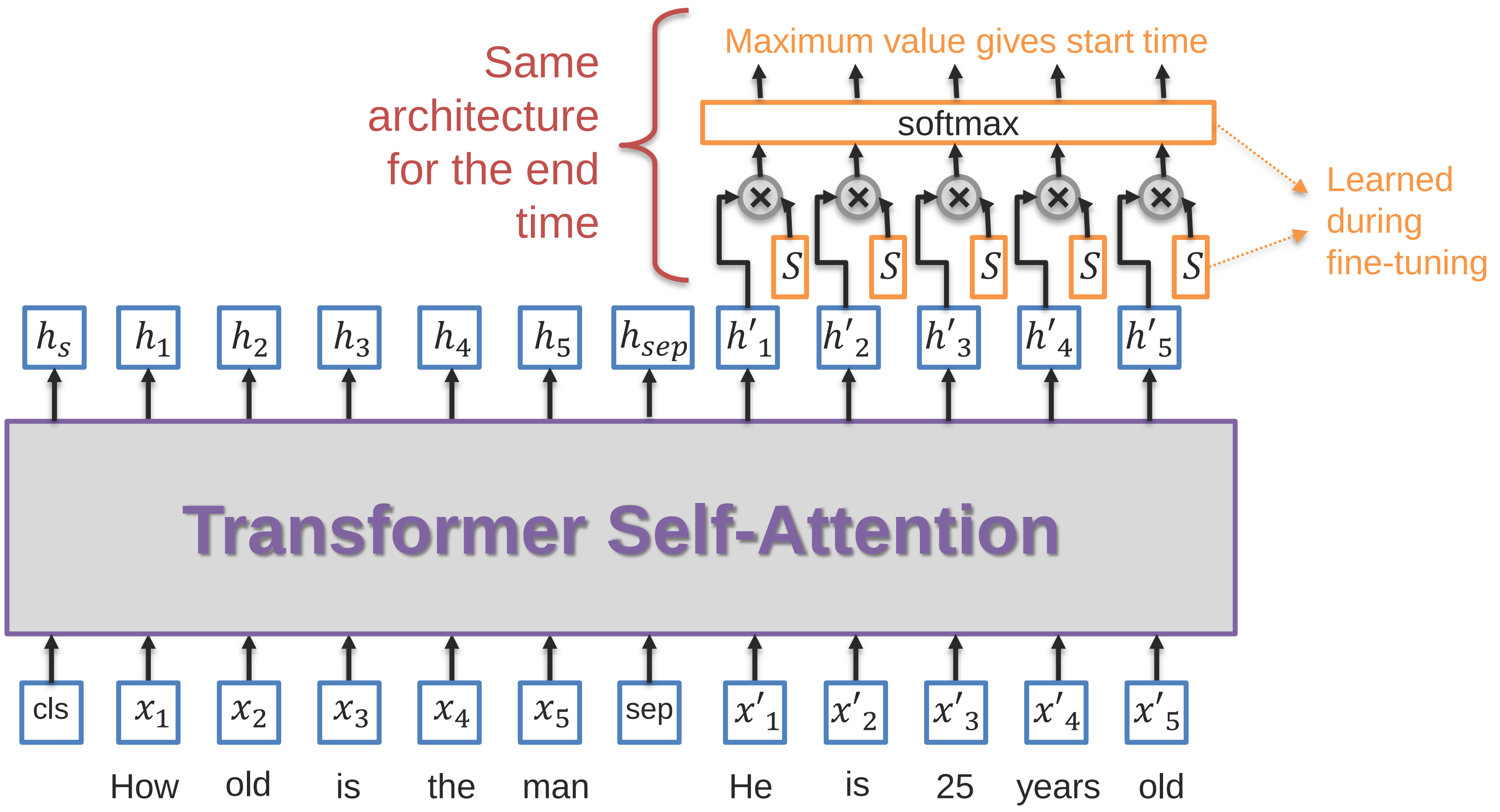
Question 2: "*What was the name of the 1937 treaty?*"
Plausible Answer: *Bald Eagle Protection Act*



How?

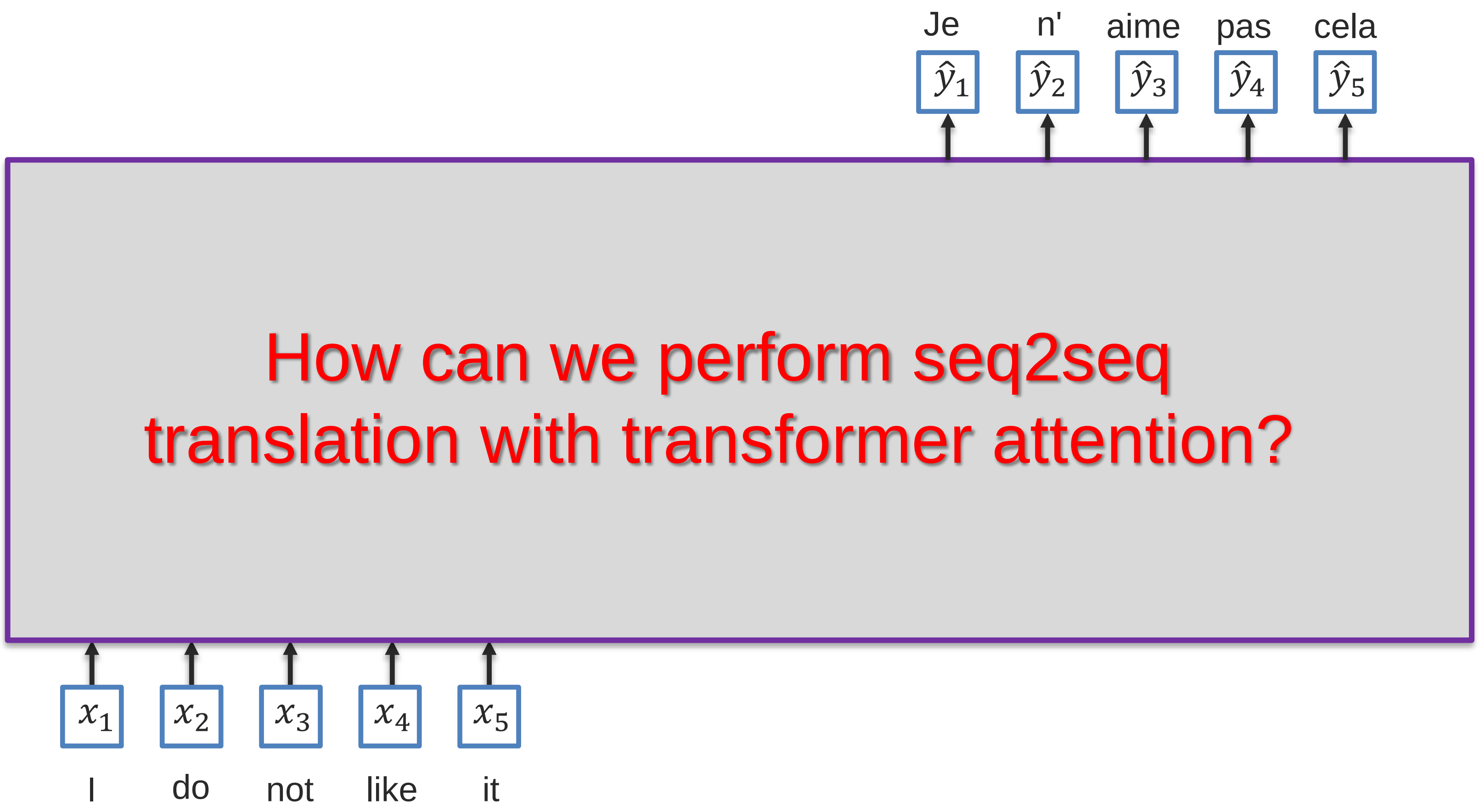
Fine-Tuning BERT

4 Question-answering: find start/end of the answer in the document

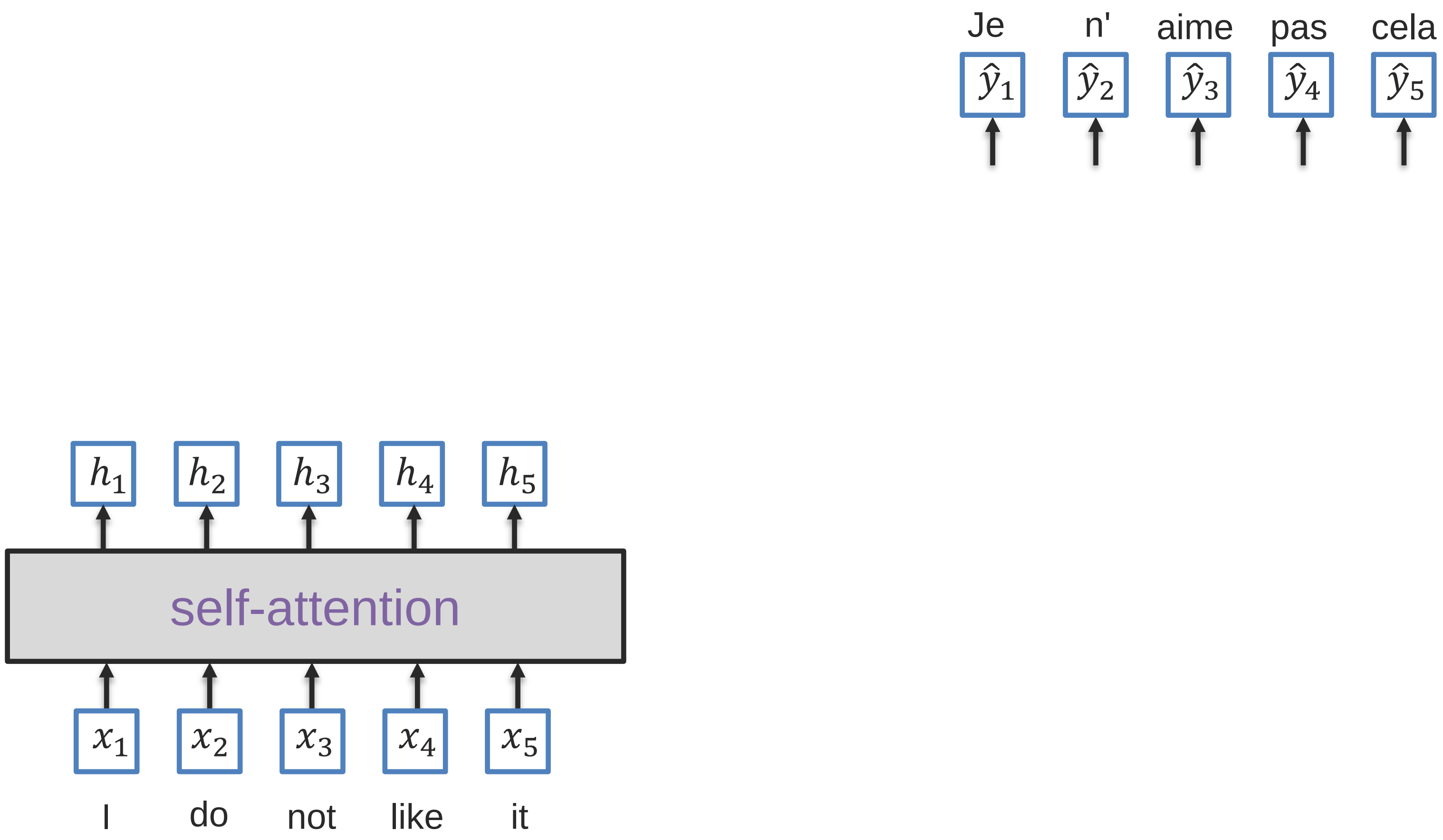


Sequence-to-Sequence Using Transformer

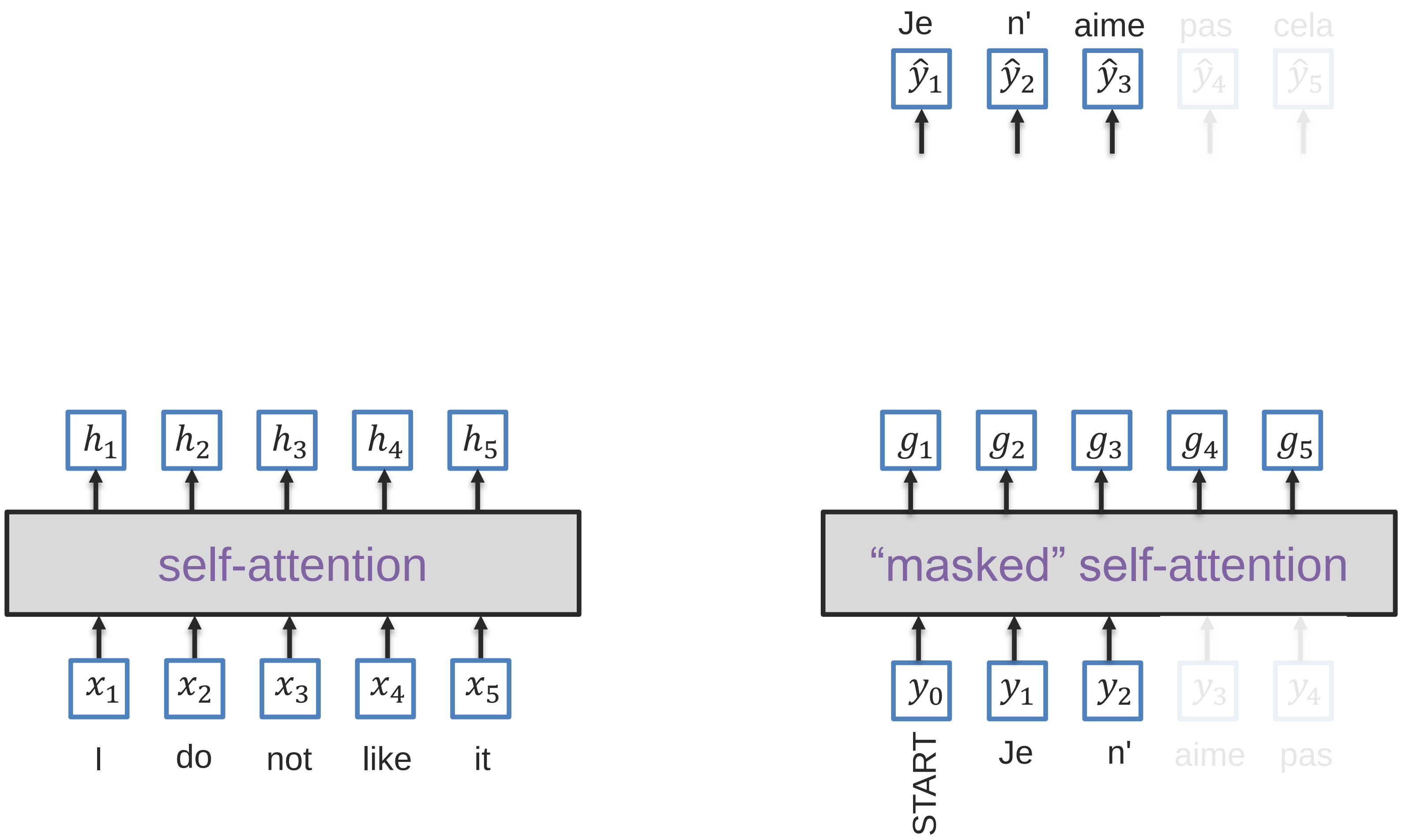
Sequence-to-Sequence Modeling



Seq2Seq with Transformer Attentions

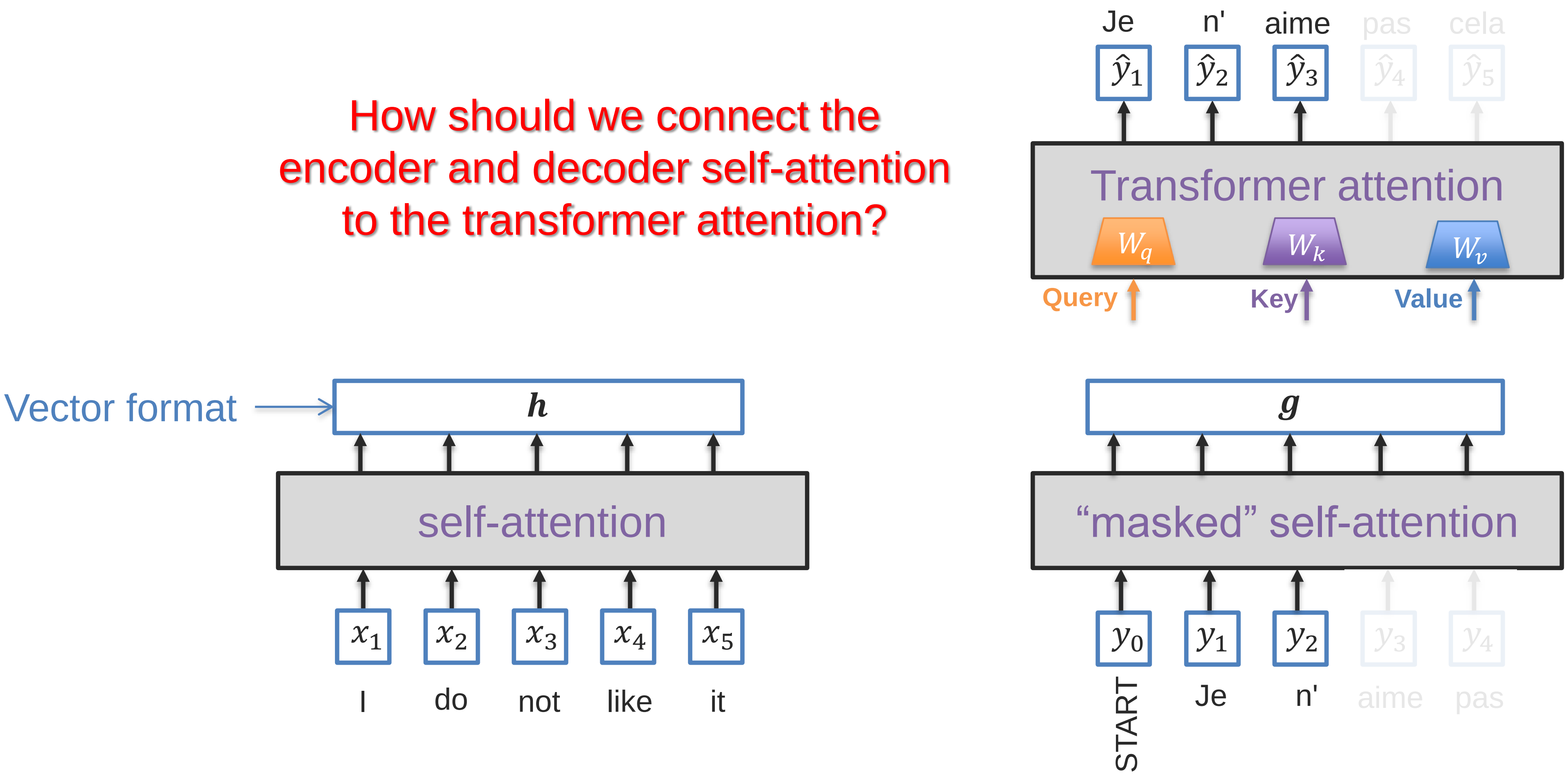


Seq2Seq with Transformer Attentions

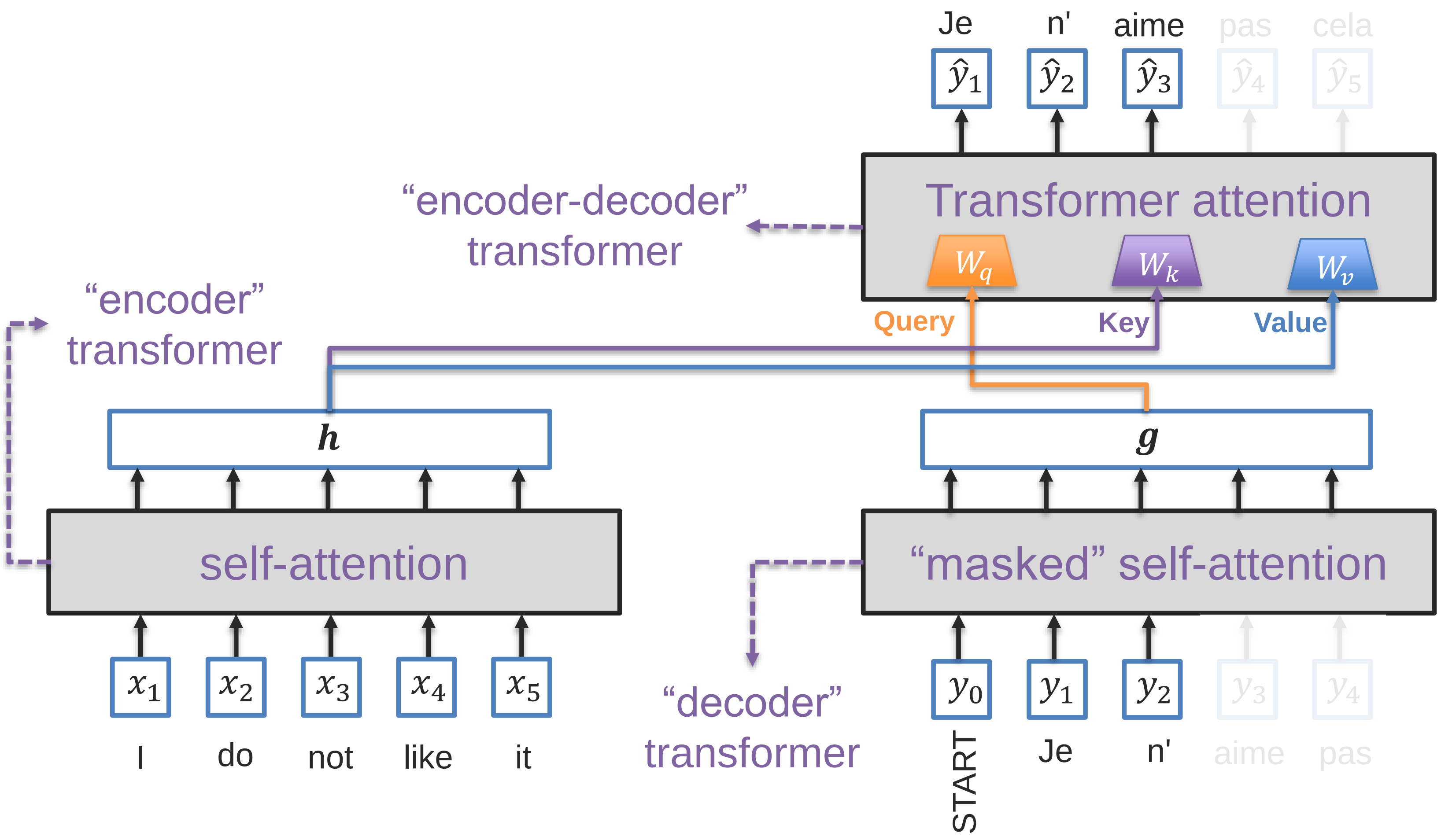


Seq2Seq with Transformer Attentions

How should we connect the encoder and decoder self-attention to the transformer attention?



Seq2Seq with Transformer Attentions



And Many More…   Next week!